PIC24FJ256GA110 系列 数据手册

具有外设引脚选择功能的 64/80/100 引脚 16 位通用闪存单片机

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下， Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是 "牢不可破"的。

代码保护功能处于持续发展中。 Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了 《数字器件千年版权法案 （Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2002 认证。公司在 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

PIC24FJ256GA110 系列

具有外设引脚选择功能的 64/80/100 引脚 16 位通用闪存单片机

功耗管理：

- 片内 2.5V 稳压器
- 实时时钟源切换
- 可快速唤醒的空闲、休眠和打盹 （Doze）模式，以及双速启动
- 运行模式：1 mA/MIPS， 2.0V （典型值）
- 使用 32 kHz 振荡器的待机模式下的电流：2.6 μA （2.0V 条件下的典型值）

高性能 CPU：

- 改进的哈佛架构
- 最高运行速度可达 16 MIPS @ 32 MHz
- 8 MHz 内部振荡器
- 17 X 17 位单周期硬件乘法器
- 32 /16 位硬件除法器
- 16 x16 位工作寄存器阵列
- 具有灵活寻址模式的优化的 C 编译器指令集架构
- 线性程序存储器的寻址范围最大 12 MB
- 线性数据存储器的寻址范围最大 64 KB
- 两个用于独立的读和写操作以寻址数据存储器的地址发生单元

模拟特性：

- 10 位、最多 16 路通道，转换速度为 500 ksps 的模数转换器 （Analog-to-Digital， A/D）：
  - 休眠模式下仍可进行转换
- 三个具有可编程输入 / 输出配置的模拟比较器
- 充电时间测量单元 （Charge Time Measurement Unit，CTMU）

外设特性：

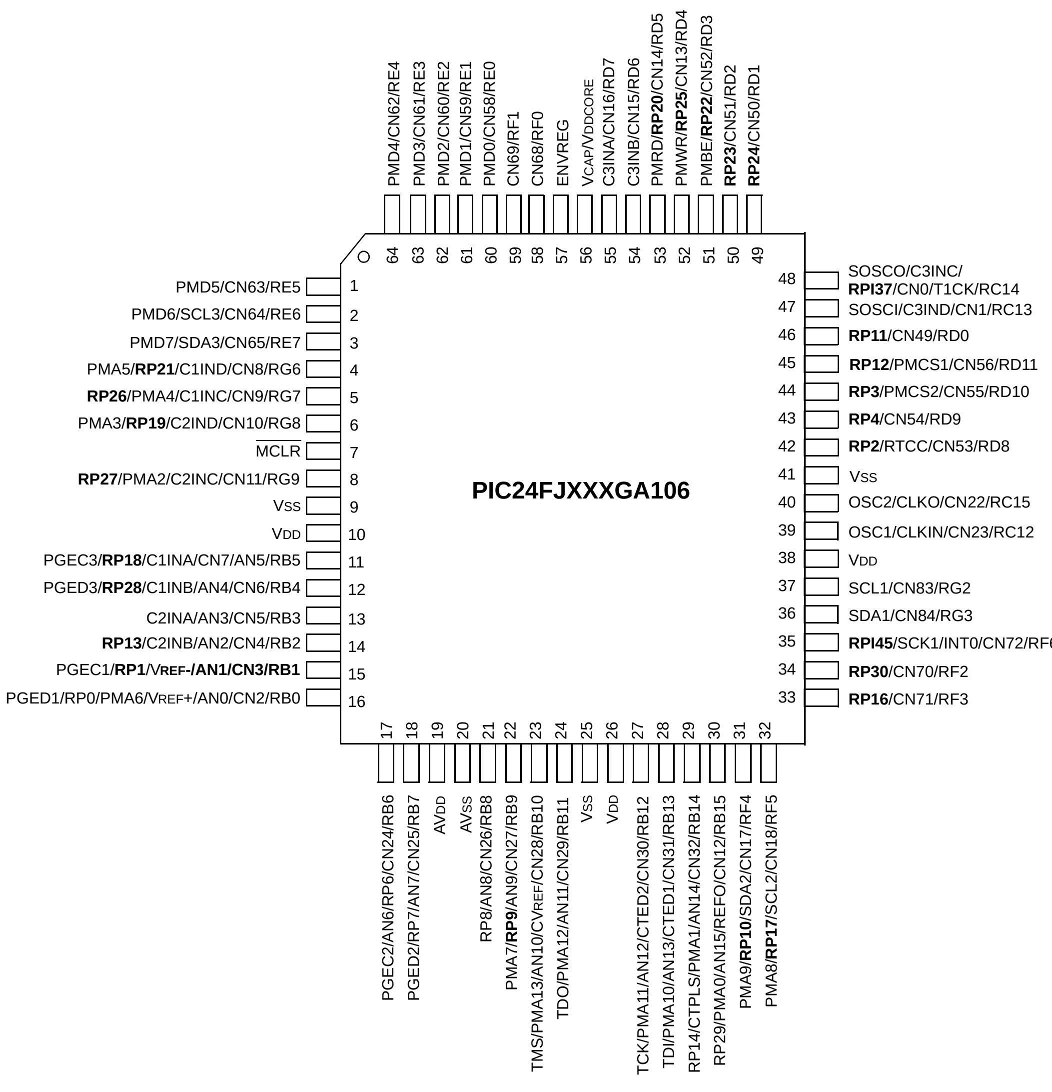
- 外设引脚选择：
  - 允许在运行时对许多外设进行独立的 I/O 映射
  - 连续的硬件完整性检查和安全互锁以防止无意中更改配置
  - 最多 44 个可用引脚 （100 引脚器件）
- 三个 3 线 /4 线 SPI 模块 （支持 4 帧模式），带 8 级 FIFO 缓冲区
- 三个 I^2C™ 模块，支持多主器件 / 从模式和 7 位 /10 位寻址
- 四个 UART 模块：
  - 支持 RS-485、 RS-232、 LIN/J6202 协议和 IrDA®
  - IrDA 使用片上硬件编码器 / 解码器
  - 自动唤醒和自动波特率检测 （Auto-Baud Detect，ABD）
  - 4 级深 FIFO 缓冲区
- 五个带可编程预分频器的 16 位定时器 / 计数器
- 九个 16 位捕捉输入，每个捕捉输入都具有一个专用时基
- 九个 16 位比较 /PWM 输出，每个比较 /PWM 输出都具有一个专用时基
- 8 位并行主端口 （Parallel Master， PMP）：
  - 最多 16 个地址引脚
  - 控制线上的可编程优先级
- 硬件实时时钟 / 日历 （Real-Time Clock/Calendar，RTCC）：
  - 提供时钟、日历和闹钟功能
- 可编程循环冗余校验 （Cyclic Redundancy Check，CRC）发生器
- 最多 5 个外部中断源

PIC24FJ 器件	引脚	程序存储器 (字节)	SRAM （字节）	可重新映射的外设: 可重映射的引脚	16 位定时器	捕捉输入	比较 / PWM 输出	带 IrDA® 的 UART	SPI	I^2C™	10 位 A/D （通道数）	比较器	PMP/PSP	JTAG	CTMU
128GA106	64	128K	16K	29	5	9	9	4	3	3	16	3	有	有	有
192GA106	64	192K	16K	29	5	9	9	4	3	3	16	3	有	有	有
256GA106	64	256K	16K	29	5	9	9	4	3	3	16	3	有	有	有
128GA108	80	128K	16K	40	5	9	9	4	3	3	16	3	有	有	有
192GA108	80	192K	16K	40	5	9	9	4	3	3	16	3	有	有	有
256GA108	80	256K	16K	40	5	9	9	4	3	3	16	3	有	有	有
128GA110	100	128K	16K	44	5	9	9	4	3	3	16	3	有	有	有
192GA110	100	192K	16K	44	5	9	9	4	3	3	16	3	有	有	有
256GA110	100	256K	16K	44	5	9	9	4	3	3	16	3	有	有	有

单片机的特殊功能：

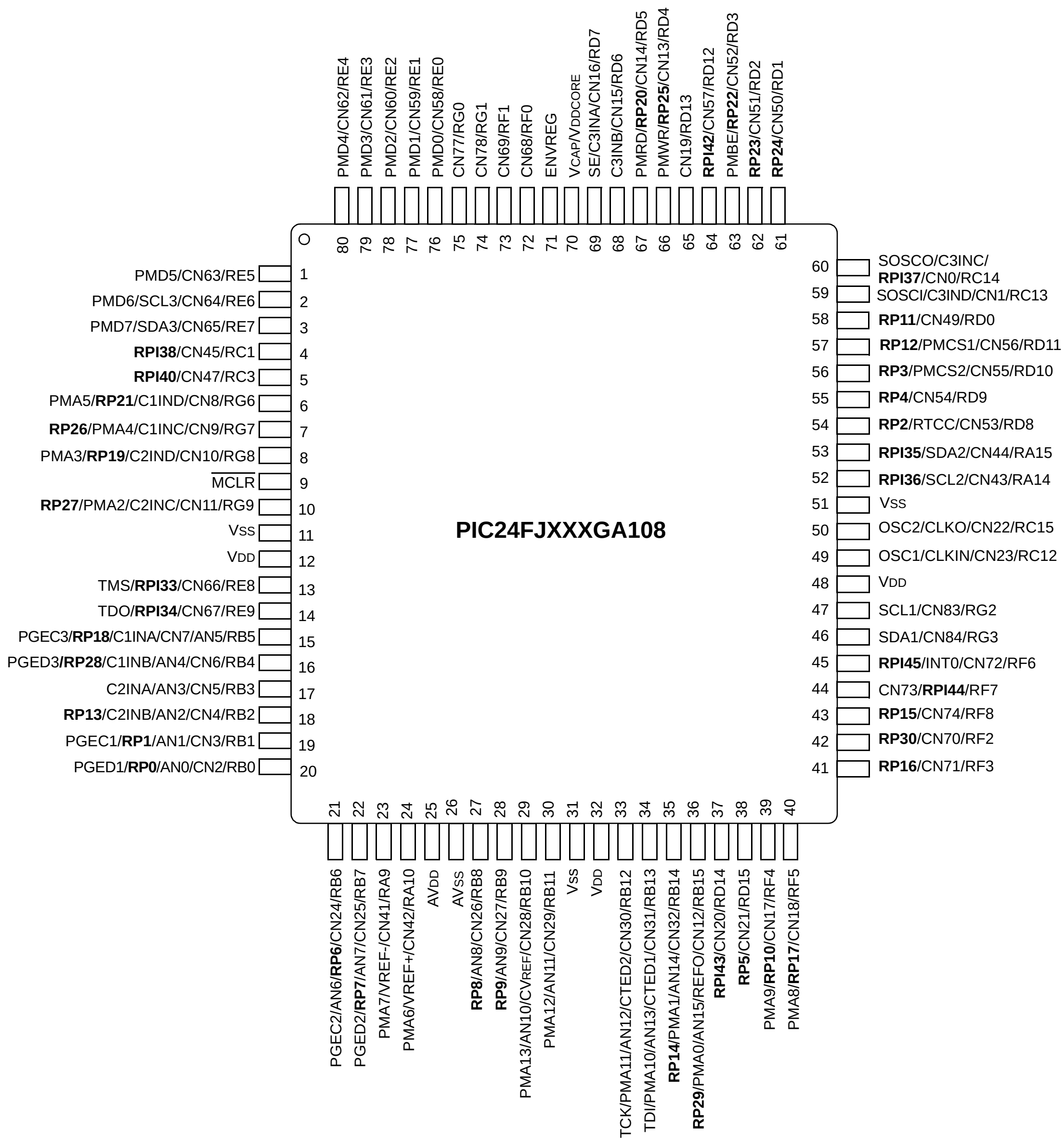
- 工作电压范围为 2.0V 至 3.6V
- 软件控制下可自行再编程
- 可承受 5.5V 输入电压 （仅数字引脚）
- 数字 I/O 引脚上的可配置漏极开路输出
- 所有 I/O 引脚上的高灌 / 拉电流 （18 mA/18 mA）
- 可选功耗管理模式：
  - 可快速唤醒的空闲、休眠和打盹模式
- 故障保护时钟监视器操作：
  - 检测时钟故障并切换至片上低功耗 RC 振荡器
- 片内 LDO 稳压器
- 上电复位 （Power-on Reset， POR）、上电延时定时器 （Power-up Timer， PWRT）、低压检测 （Low-Voltage Detect， LVD）和振荡器起振定时器 （Oscillator Start-up Timer， OST）
- 灵活的看门狗定时器 （Watchdog Timer， WDT），带有片上低功耗 RC 振荡器能够可靠工作
- 在线串行编程 （In-Circuit Serial Programming™， ICSP™）和通过 2 引脚进行的在线调试 （In-Circuit Debug， ICD）
- JTAG 边界扫描和编程支持
- 欠压复位 （Brown-out Reset， BOR）
- 闪存程序存储器：
  - 可耐受 10,000 次擦 / 写 （最少）
  - 数据保存时间最少 20 年
  - 可选的写保护边界
  - 闪存配置字的写保护选项

引脚图 （64 引脚 TQFP）



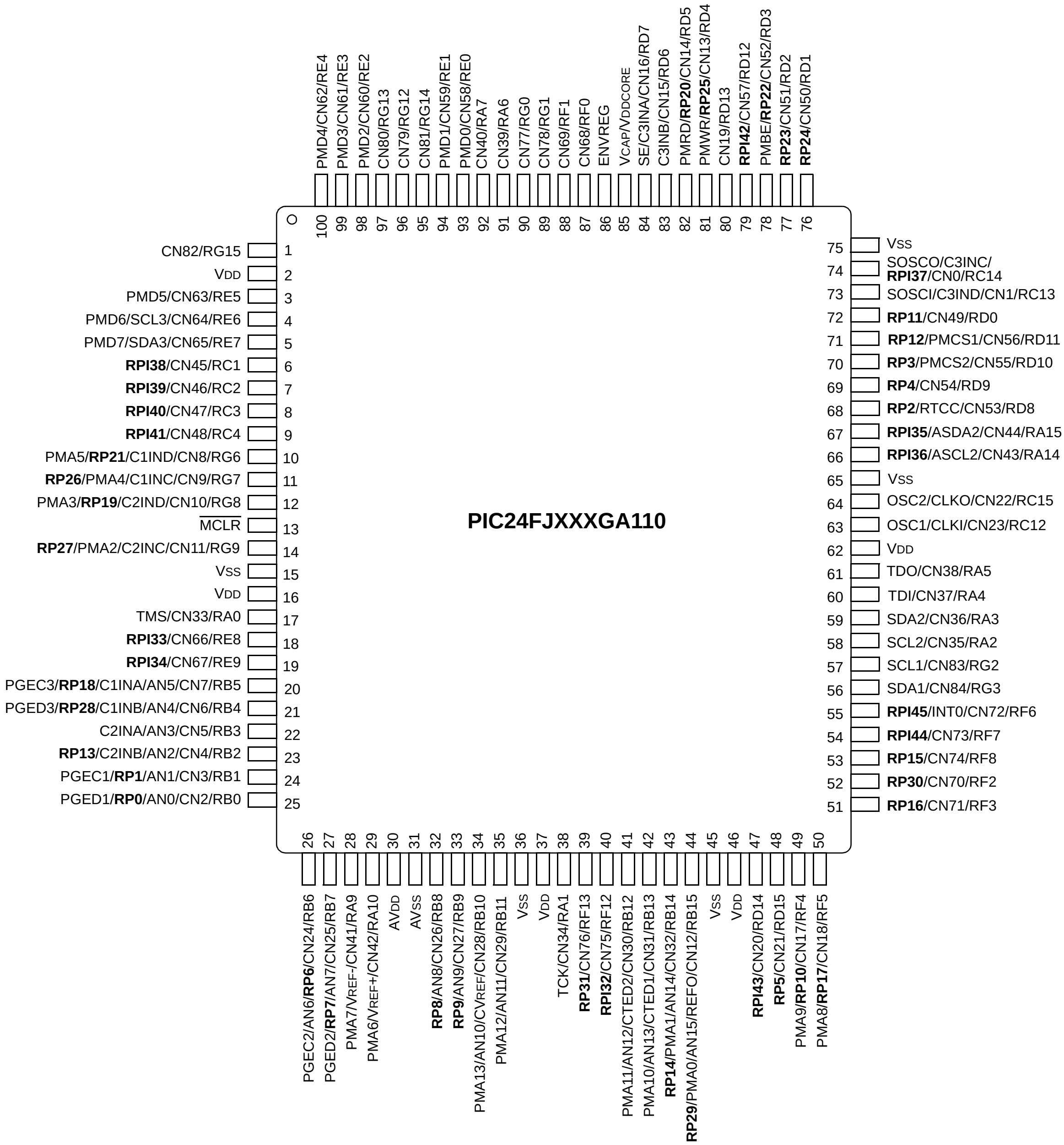
图注： **RPn** 表示用于外设引脚选择功能的可重映射引脚。

引脚图 （80 引脚 TQFP）



图注： **RPn** 表示用于外设引脚选择功能的可重映射引脚。

引脚图 （100 引脚 TQFP）



图注： **RPn** 表示用于外设引脚选择功能的可重映射引脚。

 超前信息

目录

致 客 户

我们旨在提供最佳文档供客户正确使用 Microchip 产品。为此，我们将不断改进出版物的内容和质量，使之更好地满足您的要求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 **docerrors@microchip.com**，或将本数据手册后附的 《**读者反馈表**》传真到 86-21-5407 5066。我们期待您的反馈。

最新数据手册

欲获得本数据手册的最新版本，请查询我公司的网站：

http://www.microchip.com

查看数据手册中任意一页下边角处的文献编号即可确定其版本。 文献编号中数字串后的字母是版本号，例如：DS30000A 是 DS30000 的 A 版本。

勘误表

现有器件可能带有一份勘误表，描述了实际运行与数据手册中记载内容之间存在的细微差异以及建议的变通方法。一旦我们了解到器件 / 文档存在某些差异时，就会发布勘误表。勘误表上将注明其所适用的硅片版本和文件版本。

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

- Microchip 网站 http://www.microchip.com
- 当地 Microchip 销售办事处 （见最后一页）

在联络销售办事处时，请说明您所使用的器件型号、硅片版本和数据手册版本 （包括文献编号）。

客户通知系统

请在我们的网址 **www.microchip.com** 上注册，以便及时获得我们所有产品的最新信息。

1.0 器件概述

该文档包含针对以下器件的具体信息：

- PIC24FJ128GA106
- PIC24FJ192GA106
- PIC24FJ256GA106
- PIC24FJ128GA108
- PIC24FJ192GA108
- PIC24FJ256GA108
- PIC24FJ128GA110
- PIC24FJ192GA110
- PIC24FJ256GA110

这一系列器件对 Microchips 现有的 16 位通用单片机系列进行了扩展，其中包括增强的计算性能以及扩展的可配置性极强的外设功能集。PIC24FJ256GA110 系列为那些 8 位平台无法满足其需求但又不要求使用数字信号处理器的高性能应用提供了一种新的平台。

1.1 内核特性

1.1.1 16 位架构

所有 PIC24F 器件的内核均采用了 16 位改进的哈佛架构，该架构最初是在 Microchip 的 dsPIC® 数字信号控制器中采用的。PIC24F CPU 内核提供了大量增强功能，如：

- 16 位数据路径和 24 位地址路径，可在数据空间和其他存储空间传递信息
- 线性寻址空间最多可达 12 MB （程序）和 64 KB （数据）
- 利用内建软件堆栈支持 16 元工作寄存器阵列
- 支持整数运算的 17 x 17 位硬件乘法器
- 支持 32 /16 位除法运算的硬件
- 支持多种寻址模式并为高级语言 （如 C 语言）而优化的指令集
- 工作性能最高可达 16 MIPS

1.1.2 节能技术

PIC24FJ256GA110 系列的所有器件都具有许多能在工作时显著降低功耗的功能。主要包括以下几项：

- **动态时钟切换：**在器件工作过程中，器件时钟可在软件控制下切换为 Timer1 时钟源或内部低功耗 RC 振荡器，允许用户把节能理念融入到软件设计中去。
- **打盹模式操作：**当那些对时间要求很高的应用（如串行通信）要求外设不间断地工作时，该模式可以适当降低 CPU 时钟速度，从而可在不丢失时钟的前提下进一步节约功耗。
- **基于指令的节能模式：**通过在软件中使用一条指令，单片机可以暂停所有的操作或仅关闭内核，而让外设处于活动状态。

1.1.3 振荡器选项和特性

PIC24FJ256GA110 系列中的所有器件均提供 5 个不同的振荡器选项，使用户在开发应用硬件时有很大的选择范围。这些选项包括：

- 使用晶振或陶瓷谐振器的两种晶振模式。
- 提供 2 分频时钟输出选项的两种外部时钟模式。
- 一个标称输出值为 8 MHz 的快速内部振荡器（Fast Internal Oscillator， FRC），可以在软件控制下被分频，从而使时钟速度可低至 31 kHz。
- 一个锁相环 （Phase Lock Loop， PLL）倍频器，可在外部振荡器模式和采用 FRC 振荡器的情况下使用，从而使时钟速度最高可达 32 MHz。
- 具有固定的 31 kHz 输出的独立内部 RC 振荡器（LPRC），可为对时间要求不高的应用提供低功耗时钟选项。

内部振荡器模块还为故障保护时钟监视器提供了一个稳定的参考源。故障保护时钟监视器不断地监视主时钟源，将之与内部振荡器提供的参考信号作比较。一旦发生时钟故障，允许控制器将时钟源切换到内部振荡器，继续保持低速工作或安全地关闭应用。

1.1.4 易于移植

无论存储器大小如何，所有器件均共享同一组外设，使得应用程序可在升级时很方便地移植。整个系列使用相同的引脚配置方案也有助于向更大型器件的移植，甚至还可以从 64 引脚器件移植到 100 引脚的器件。

PIC24F 系列器件的引脚同 dsPIC33 系列器件的引脚是兼容的，并与 PIC18 和 dsPIC30 的引脚部分兼容。这样全部采用 Microchip 器件，就可将应用从相对简单的功能顺利扩展到强大和复杂的功能。

1.2 其他特性

- **外设引脚选择：**外设引脚选择功能允许大部分的数字外设被映射到固定的数字 I/O 引脚集。用户可独立地将众多数字外设之一的输入和 / 或输出映射到其中的任一 I/O 引脚。
- **通信：**PIC24FJ256GA110 系列集成了一些串行通信外设以满足一系列的应用要求。有三个独立的 I^2C 模块支持主模式和从模式下的操作。通过外设引脚选择功能，器件还具有四个带内建 IrDA 编 / 解码器的独立 UART 以及三个 SPI 模块。
- **模拟特性：**PIC24FJ256GA110 系列的所有组成部分包括一个 10 位的 A/D 转换模块和一个三比较器模块。 A/D 转换模块实现了可编程采集时间，允许选择通道并立即开始转换而无需等待采样周期结束，同时也提高了采样速度。比较器模块包含三个模拟比较器，它们可配置为多种工作模式。
- **CTMU 接口：**除了其他模拟特性外，PIC24FJ256GA110 系列的组成部分还包含了一种全新的 CTMU 接口模块。这为精确时间测量和脉冲产生提供了一种便捷的方法，同时也可以作为容性传感器的接口。
- **并行主 / 增强型并行从端口：**可以将一个通用 I/O 端口重新配置为用于增强型并行数据通信。在这种模式下，可以将端口配置为工作在主或从模式下。在主模式下支持 8 位或 16 位数据传输并具有最多 16 条外部地址线。
- **实时时钟 / 日历：**此模块通过硬件实现带有闹钟功能的全功能时钟和日历，从而释放了定时器资源和程序存储空间供核心应用使用。

1.3 系列中各产品的具体信息

PIC24FJ256GA110 系列中的器件有 64 引脚、 80 引脚和 100 引脚封装。所有器件的一般框图如图 1-1 所示。

这些器件在以下四个方面存在差异：

1. 闪存程序存储器（PIC24FJ128GA1器件为128 KB、PIC24FJ192GA1 器件为 192 KB 而 PIC24FJ256GA1 器件为 256 KB）。
2. 可用的 I/O 引脚和端口数（64 引脚的器件有 6 个端口 53 个引脚， 80 引脚的器件有 7 个端口 69 个引脚而100引脚的器件有7个端口85个引脚）。
3. 可用的电平变化中断通知（Interrupt-on-Change Notification， ICN）输入的数量与各器件可用的 I/O 引脚数相同。
4. 可用的可重新映射引脚（64 引脚的器件有 31 个引脚，80 引脚的器件有 42 个引脚而 100 引脚的器件有 46 个引脚）。

该系列器件的其他特性都是相同的。表 1-1 对此进行了总结。

PIC24FJ256GA110 系列器件上可用的引脚功能列表如表 1-4 所示，按功能名称排序。注意此表只显示了各个外设功能所使用的引脚，而没有显示同一引脚上的多种功能的复用方式。在本数据手册开始部分的引脚图中提供了相关信息。复用的功能按功能的优先级排列，最前面的是优先级最高的外设功能。

表 **1-1**： **PIC24FJ256GA110** 系列的器件特性：**64** 引脚器件

特性	128GA106	192GA106	256GA106
工作频率	DC – 32 MHz		
程序存储器 (字节)	128K	192K	256K
程序存储器 (指令)	44,032	67,072	87,552
数据存储器 (字节)	16,384		
中断源 (软向量 /NMI 陷阱)	66(62/4)		
I/O 端口	B、 C、 D、 E、 F 和 G 端口		
I/O 引脚总数	53		
可重映射引脚	共 31 个 (29 个 I/O 引脚和 2 个仅用作输入的引脚)		
定时器： 总数 (16 位)	5(1)		
32 位 (由一对 16 位定时器组成)	2		
输入捕捉通道	9(1)		
输出比较 /PWM 通道	9(1)		
输入变化通知中断	53		
串行通信： UART	4(1)		
SPI(3 线 /4 线)	3(1)		
I^2C™	3		
并行通信 (PMP/PSP)	有		
JTAG 边界扫描 / 编程	有		
10 位模数转换模块 (输入通道)	16		
模拟比较器	3		
CTMU 接口	有		
复位 (和延时)	POR、 BOR、 RESET 指令、 $\overline{\text{MCLR}}$、 WDT ；非法操作码、 REPEAT 指令、硬件陷阱和配置字失配 (PWRT、 OST 和 PLL 锁定)		
指令集	76 条基本指令和多种寻址模式		
封装	64 引脚 TQFP 封装		

注 **1**： 可通过可重新映射的引脚访问外设。

表 1-2： PIC24FJ256GA110 系列器件的特性：80 引脚器件

特性	128GA108	192GA108	256GA108
工作频率	DC – 32 MHz		
程序存储器 (字节)	128K	192K	256K
程序存储器 (指令)	44,032	67,072	87,552
数据存储器 (字节)	16,384		
中断源 (软向量 /NMI 陷阱)	66(62/4)		
I/O 端口	A、 B、 C、 D、 E、 F 和 G 端口		
I/O 引脚总数	69		
可重映射引脚	共 42 个 (31 个 I/O 引脚和 11 个仅用作输入的引脚)		
定时器： 总数 (16 位)	5(1)		
32 位 (由一对 16 位定时器组成)	2		
输入捕捉通道	9(1)		
输出比较 /PWM 通道	9(1)		
输入变化通知中断	69		
串行通信： UART	4(1)		
SPI(3 线 /4 线)	3(1)		
I^2C™	3		
并行通信 (PMP/PSP)	有		
JTAG 边界扫描 / 编程	有		
10 位模数转换模块 (输入通道)	16		
模拟比较器	3		
CTMU 接口	有		
复位 (和延时)	POR、 BOR、 RESET 指令、 $\overline{\text{MCLR}}$、 WDT；非法操作码、 REPEAT 指令、硬件陷阱和配置字失配 (PWRT、 OST 和 PLL 锁定)		
指令集	76 条基本指令和多种寻址模式		
封装	80 引脚 TQFP 封装		

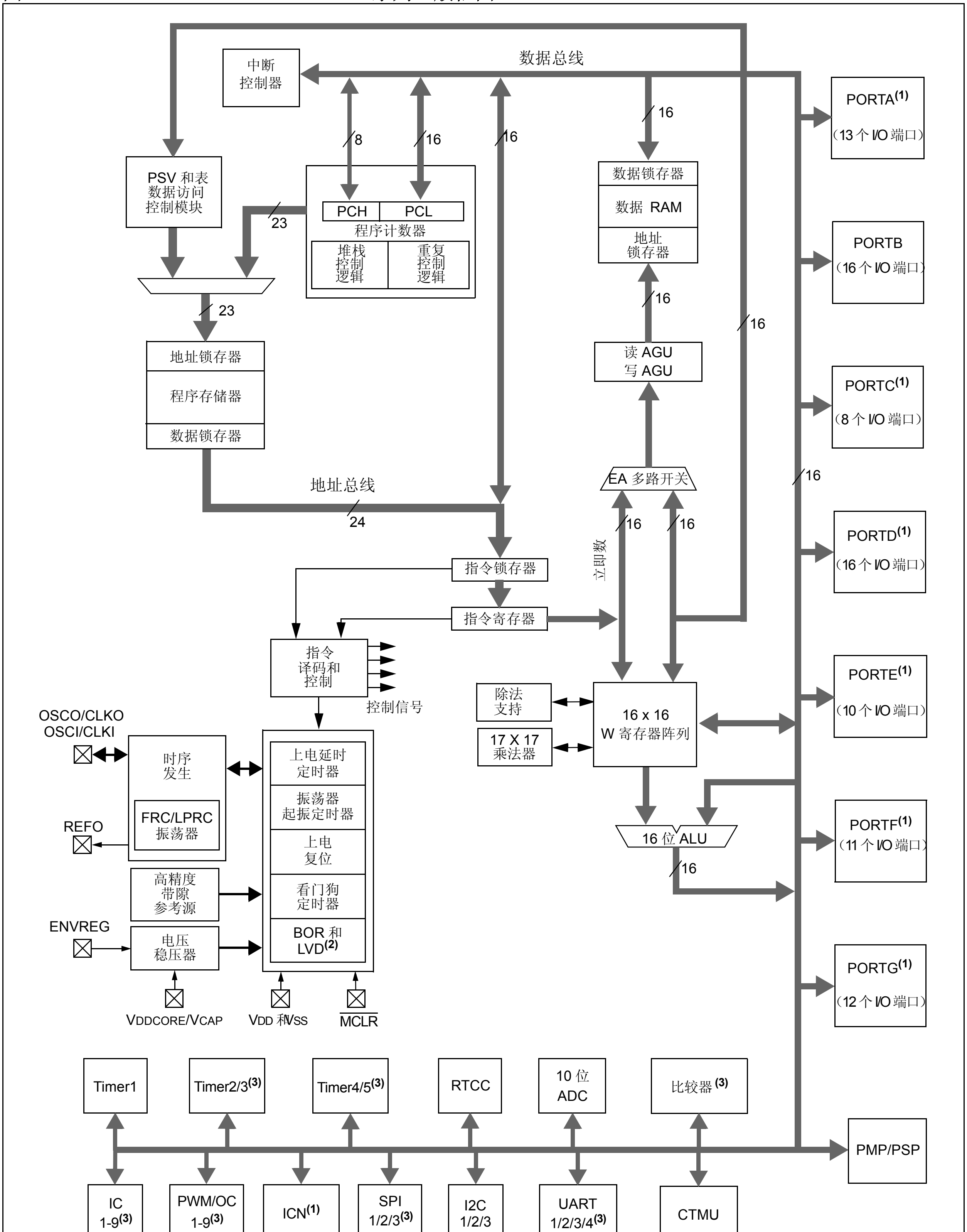
注 1： 可通过可重新映射的引脚访问外设。

表 1-3： PIC24FJ256GA110 系列器件的特性：100 引脚器件

特性	128GA110	192GA110	256GA110
工作频率	DC – 32 MHz		
程序存储器 (字节)	128K	192K	256K
程序存储器 (指令)	44,032	67,072	87,552
数据存储器 (字节)	16,384		
中断源 (软向量 /NMI 陷阱)	66(62/4)		
I/O 端口	A、 B、 C、 D、 E、 F 和 G 端口		
I/O 引脚总数	85		
可重映射引脚	共 46 个 (32 个 I/O 引脚和 14 个仅用作输入的引脚)		
定时器： 总数 (16 位)	5(1)		
32 位 (由一对 16 位定时器组成)	2		
输入捕捉通道	9(1)		
输出比较 /PWM 通道	9(1)		
输入变化通知中断	85		
串行通信： UART	4(1)		
SPI(3 线 /4 线)	3(1)		
I^2C™	3		
并行通信 (PMP/PSP)	有		
JTAG 边界扫描 / 编程	有		
10 位模数转换模块 (输入通道)	16		
模拟比较器	3		
CTMU 接口	有		
复位 (和延时)	POR、 BOR、 RESET 指令、 $\overline{\text{MCLR}}$、 WDT ；非法操作码、 REPEAT 指令、硬件陷阱和配置字失配 (PWRT、 OST 和 PLL 锁定)		
指令集	76 条基本指令和多种寻址模式		
封装	100 引脚 TQFP 封装		

注 1： 可通过可重新映射的引脚访问外设。

图 1-1： PIC24FJ256GA110 系列一般框图



注 1：并非所有引脚或特性都能在所有器件的引脚配置中实现。实现的具体引脚数，请参见表 1-4。
2：当使能片上稳压器时可提供 BOR 功能。
3：只可通过可重映射的引脚访问外设 I/O。

表 **1-4：** **PIC24FJ256GA110** 系列引脚配置说明

功能	引脚号			I/O	输入缓冲器	说明
	64 引脚 TQFP 封装	**80 引脚 TQFP 封装**	**100 引脚 TQFP 封装**			
AN0	16	20	25	I	ANA	A/D 模拟输入。
AN1	15	19	24	I	ANA	
AN2	14	18	23	I	ANA	
AN3	13	17	22	I	ANA	
AN4	12	16	21	I	ANA	
AN5	11	15	20	I	ANA	
AN6	17	21	26	I	ANA	
AN7	18	22	27	I	ANA	
AN8	21	27	32	I	ANA	
AN9	22	28	33	I	ANA	
AN10	23	29	34	I	ANA	
AN11	24	30	35	I	ANA	
AN12	27	33	41	I	ANA	
AN13	28	34	42	I	ANA	
AN14	29	35	43	I	ANA	
AN15	30	36	44	I	ANA	
ASCL2	—	—	66	I/O	I^2C	备用 I2C2 同步串行时钟输入 / 输出。
ASDA2	—	—	67	I/O	I^2C	备用 I2C2 数据输入 / 输出。
AVDD	19	25	30	P	—	模拟模块的正电源。
AVSS	20	26	31	P	—	模拟模块的接地参考端。
C1INA	11	15	20	I	ANA	比较器 1 输入 A。
C1INB	12	16	21	I	ANA	比较器 1 输入 B。
C1INC	5	7	11	I	ANA	比较器 1 输入 C。
C1IND	4	6	10	I	ANA	比较器 1 输入 D。
C2INA	13	17	22	I	ANA	比较器 2 输入 A。
C2INB	14	18	23	I	ANA	比较器 2 输入 B。
C2INC	8	10	14	I	ANA	比较器 2 输入 C。
C2IND	6	8	12	I	ANA	比较器 2 输入 D。
C3INA	55	69	84	I	ANA	比较器 3 输入 A。
C3INB	54	68	83	I	ANA	比较器 3 输入 B。
C3INC	48	60	74	I	ANA	比较器 3 输入 C。
C3IND	47	59	73	I	ANA	比较器 3 输入 D。
CLKI	39	49	63	I	ANA	主时钟输入连接。
CLKO	40	50	64	O	—	系统时钟输出。

图注： TTL = TTL 输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电平输入 / 输出　　I^2C™ = I^2C/SMBus 输入缓冲器

表 1-4： PIC24FJ256GA110 系列引脚配置说明 （续）

功能	引脚号 64 引脚 TQFP 封装	引脚号 80 引脚 TQFP 封装	引脚号 100 引脚 TQFP 封装	I/O	输入缓冲器	说明
CN0	48	60	74	I	ST	电平变化中断输入。
CN1	47	59	73	I	ST	
CN2	16	20	25	I	ST	
CN3	15	19	24	I	ST	
CN4	14	18	23	I	ST	
CN5	13	17	22	I	ST	
CN6	12	16	21	I	ST	
CN7	11	15	20	I	ST	
CN8	4	6	10	I	ST	
CN9	5	7	11	I	ST	
CN10	6	8	12	I	ST	
CN11	8	10	14	I	ST	
CN12	30	36	44	I	ST	
CN13	52	66	81	I	ST	
CN14	53	67	82	I	ST	
CN15	54	68	83	I	ST	
CN16	55	69	84	I	ST	
CN17	31	39	49	I	ST	
CN18	32	40	50	I	ST	
CN19	—	65	80	I	ST	
CN20	—	37	47	I	ST	
CN21	—	38	48	I	ST	
CN22	40	50	64	I	ST	
CN23	39	49	63	I	ST	
CN24	17	21	26	I	ST	
CN25	18	22	27	I	ST	
CN26	21	27	32	I	ST	
CN27	22	28	33	I	ST	
CN28	23	29	34	I	ST	
CN29	24	30	35	I	ST	
CN30	27	33	41	I	ST	
CN31	28	34	42	I	ST	
CN32	29	35	43	I	ST	
CN33	—	—	17	I	ST	
CN34	—	—	38	I	ST	
CN35	—	—	58	I	ST	
CN36	—	—	59	I	ST	
CN37	—	—	60	I	ST	
CN38	—	—	61	I	ST	
CN39	—	—	91	I	ST	
CN40	—	—	92	I	ST	
CN41	—	23	28	I	ST	
CN42	—	24	29	I	ST	

图注： TTL = TTL 输入缓冲器 ST = 施密特触发器输入缓冲器
ANA = 模拟电平输入 / 输出 I^2C™ = I^2C/SMBus 输入缓冲器

表 1-4： PIC24FJ256GA110 系列引脚配置说明 （续）

功能	引脚号			I/O	输入缓冲器	说明
	64 引脚 TQFP 封装	80 引脚 TQFP 封装	100 引脚 TQFP 封装			
CN43	—	52	66	I	ST	电平变化中断输入。
CN44	—	53	67	I	ST	
CN45	—	4	6	I	ST	
CN46	—	—	7	I	ST	
CN47	—	5	8	I	ST	
CN48	—	—	9	I	ST	
CN49	46	58	72	I	ST	
CN50	49	61	76	I	ST	
CN51	50	62	77	I	ST	
CN52	51	63	78	I	ST	
CN53	42	54	68	I	ST	
CN54	43	55	69	I	ST	
CN55	44	56	70	I	ST	
CN56	45	57	71	I	ST	
CN57	—	64	79	I	ST	
CN58	60	76	93	I	ST	
CN59	61	77	94	I	ST	
CN60	62	78	98	I	ST	
CN61	63	79	99	I	ST	
CN62	64	80	100	I	ST	
CN63	1	1	3	I	ST	
CN64	2	2	4	I	ST	
CN65	3	3	5	I	ST	
CN66	—	13	18	I	ST	
CN67	—	14	19	I	ST	
CN68	58	72	87	I	ST	
CN69	59	73	88	I	ST	
CN70	34	42	52	I	ST	
CN71	33	41	51	I	ST	
CN72	35	45	55	I	ST	
CN73	—	44	54	I	ST	
CN74	—	43	53	I	ST	
CN75	—	—	40	I	ST	
CN76	—	—	39	I	ST	
CN77	—	75	90	I	ST	
CN78	—	74	89	I	ST	
CN79	—	—	96	I	ST	
CN80	—	—	97	I	ST	
CN81	—	—	95	I	ST	
CN82	—	—	1	I	ST	
CN83	37	47	57	I	ST	
CN84	36	46	56	I	ST	

图注： TTL = TTL 输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电平输入 / 输出　　I^2C™ = I^2C/SMBus 输入缓冲器

表 1-4： PIC24FJ256GA110 系列引脚配置说明 （续）

功能	引脚号			I/O	输入缓冲器	说明
	64 引脚 TQFP 封装	80 引脚 TQFP 封装	100 引脚 TQFP 封装			
CTED1	28	34	42	I	ANA	CTMU 外部边沿输入 1。
CTED2	27	33	41	I	ANA	CTMU 外部边沿输入 2。
CTPLS	29	35	43	O	—	CTMU 脉冲输出。
CVREF	23	29	34	O	—	比较器参考电压输出。
ENVREG	57	71	86	I	ST	使能稳压器。
INT0	35	45	55	I	ST	外部中断输入。
$\overline{\text{MCLR}}$	7	9	13	I	ST	主复位 （器件复位）输入。将此引脚拉为低电平可导致器件复位。
OSCI	39	49	63	I	ANA	主振荡器输入连接。
OSCO	40	50	64	O	ANA	主振荡器输出连接。
PGEC1	15	19	24	I/O	ST	在线调试器 / 仿真器 / ICSP™ 编程时钟。
PGED1	16	20	25	I/O	ST	在线调试器 / 仿真器 / ICSP 编程数据。
PGEC2	17	21	26	I/O	ST	在线调试器 / 仿真器 / ICSP 编程时钟。
PGED2	18	22	27	I/O	ST	在线调试器 / 仿真器 / ICSP 编程数据。
PGEC3	11	15	20	I/O	ST	在线调试器 / 仿真器 / ICSP 编程时钟。
PGED3	12	16	21	I/O	ST	在线调试器 / 仿真器 / ICSP 编程数据。
PMA0	30	36	44	I/O	ST	并行主端口地址 Bit 0 输入 (缓冲从模式) 和输出 (主模式)。
PMA1	29	35	43	I/O	ST	并行主端口地址 Bit 1 输入 (缓冲从模式) 和输出 (主模式)。
PMA2	8	10	14	O	—	并行主端口地址 (非复用的主模式)。
PMA3	6	8	12	O	—	
PMA4	5	7	11	O	—	
PMA5	4	6	10	O	—	
PMA6	16	24	29	O	—	
PMA7	22	23	28	O	—	
PMA8	32	40	50	O	—	
PMA9	31	39	49	O	—	
PMA10	28	34	42	O	—	
PMA11	27	33	41	O	—	
PMA12	24	30	35	O	—	
PMA13	23	29	34	O	—	
PMCS1	45	57	71	I/O	ST/TTL	并行主端口片选 1 选通 / 地址 Bit 15。
PMCS2	44	56	70	O	ST	并行主端口片选 2 选通 / 地址 Bit 14。
PMBE	51	63	78	O	—	并行主端口字节使能选通。
PMD0	60	76	93	I/O	ST/TTL	并行主端口数据 （非复用的主模式）或地址 / 数据 （复用的主模式）。
PMD1	61	77	94	I/O	ST/TTL	
PMD2	62	78	98	I/O	ST/TTL	
PMD3	63	79	99	I/O	ST/TTL	
PMD4	64	80	100	I/O	ST/TTL	
PMD5	1	1	3	I/O	ST/TTL	
PMD6	2	2	4	I/O	ST/TTL	
PMD7	3	3	5	I/O	ST/TTL	
PMRD	53	67	82	O	—	并行主端口读选通。
PMWR	52	66	81	O	—	并行主端口写选通。

图注： TTL = TTL 输入缓冲器 ST = 施密特触发器输入缓冲器
ANA = 模拟电平输入 / 输出 I^2C™ = I^2C/SMBus 输入缓冲器

表 1-4： PIC24FJ256GA110 系列引脚配置说明 （续）

功能	引脚号			I/O	输入缓冲器	说明
	64 引脚 TQFP 封装	80 引脚 TQFP 封装	100 引脚 TQFP 封装			
RA0	—	—	17	I/O	ST	PORTA 数字 I/O。
RA1	—	—	38	I/O	ST	
RA2	—	—	58	I/O	ST	
RA3	—	—	59	I/O	ST	
RA4	—	—	60	I/O	ST	
RA5	—	—	61	I/O	ST	
RA6	—	—	91	I/O	ST	
RA7	—	—	92	I/O	ST	
RA9	—	23	28	I/O	ST	
RA10	—	24	29	I/O	ST	
RA14	—	52	66	I/O	ST	
RA15	—	53	67	I/O	ST	
RB0	16	20	25	I/O	ST	PORTB 数字 I/O。
RB1	15	19	24	I/O	ST	
RB2	14	18	23	I/O	ST	
RB3	13	17	22	I/O	ST	
RB4	12	16	21	I/O	ST	
RB5	11	15	20	I/O	ST	
RB6	17	21	26	I/O	ST	
RB7	18	22	27	I/O	ST	
RB8	21	27	32	I/O	ST	
RB9	22	28	33	I/O	ST	
RB10	23	29	34	I/O	ST	
RB11	24	30	35	I/O	ST	
RB12	27	33	41	I/O	ST	
RB13	28	34	42	I/O	ST	
RB14	29	35	43	I/O	ST	
RB15	30	36	44	I/O	ST	
RC1	—	4	6	I/O	ST	PORTC 数字 I/O。
RC2	—	—	7	I/O	ST	
RC3	—	5	8	I/O	ST	
RC4	—	—	9	I/O	ST	
RC12	39	49	63	I/O	ST	
RC13	47	59	73	I/O	ST	
RC14	48	60	74	I/O	ST	
RC15	40	50	64	I/O	ST	

图注： TTL = TTL 输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电平输入 / 输出　　I^2C™ = I^2C/SMBus 输入缓冲器

表 1-4： PIC24FJ256GA110 系列引脚配置说明 （续）

功能	引脚号			I/O	输入缓冲器	说明
	64 引脚 TQFP 封装	80 引脚 TQFP 封装	100 引脚 TQFP 封装			
RD0	46	58	72	I/O	ST	PORTD 数字 I/O。
RD1	49	61	76	I/O	ST	
RD2	50	62	77	I/O	ST	
RD3	51	63	78	I/O	ST	
RD4	52	66	81	I/O	ST	
RD5	53	67	82	I/O	ST	
RD6	54	68	83	I/O	ST	
RD7	55	69	84	I/O	ST	
RD8	42	54	68	I/O	ST	
RD9	43	55	69	I/O	ST	
RD10	44	56	70	I/O	ST	
RD11	45	57	71	I/O	ST	
RD12	—	64	79	I/O	ST	
RD13	—	65	80	I/O	ST	
RD14	—	37	47	I/O	ST	
RD15	—	38	48	I/O	ST	
RE0	60	76	93	I/O	ST	PORTE 数字 I/O。
RE1	61	77	94	I/O	ST	
RE2	62	78	98	I/O	ST	
RE3	63	79	99	I/O	ST	
RE4	64	80	100	I/O	ST	
RE5	1	1	3	I/O	ST	
RE6	2	2	4	I/O	ST	
RE7	3	3	5	I/O	ST	
RE8	—	13	18	I/O	ST	
RE9	—	14	19	I/O	ST	
REFO	30	36	44	O	—	参考时钟输出。
RF0	58	72	87	I/O	ST	PORTF 数字 I/O。
RF1	59	73	88	I/O	ST	
RF2	34	42	52	I/O	ST	
RF3	33	41	51	I/O	ST	
RF4	31	39	49	I/O	ST	
RF5	32	40	50	I/O	ST	
RF6	35	45	55	I/O	ST	
RF7	—	44	54	I/O	ST	
RF8	—	43	53	I/O	ST	
RF12	—	—	40	I/O	ST	
RF13	—	—	39	I/O	ST	

图注： TTL = TTL 输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电平输入 / 输出　　I^2C™ = I^2C/SMBus 输入缓冲器

 超前信息

表 1-4： PIC24FJ256GA110 系列引脚配置说明 （续）

功能	引脚号			I/O	输入缓冲器	说明
	64 引脚 TQFP 封装	80 引脚 TQFP 封装	100 引脚 TQFP 封装			
RG0	—	75	90	I/O	ST	PORTG 数字 I/O。
RG1	—	74	89	I/O	ST	
RG2	37	47	57	I/O	ST	
RG3	36	46	56	I/O	ST	
RG6	4	6	10	I/O	ST	
RG7	5	7	11	I/O	ST	
RG8	6	8	12	I/O	ST	
RG9	8	10	14	I/O	ST	
RG12	—	—	96	I/O	ST	
RG13	—	—	97	I/O	ST	
RG14	—	—	95	I/O	ST	
RG15	—	—	1	I/O	ST	
RP0	16	20	25	I/O	ST	可重映射的外设 (输入或输出)。
RP1	15	19	24	I/O	ST	
RP2	42	54	68	I/O	ST	
RP3	44	56	70	I/O	ST	
RP4	43	55	69	I/O	ST	
RP5	—	38	48	I/O	ST	
RP6	17	21	26	I/O	ST	
RP7	18	22	27	I/O	ST	
RP8	21	27	32	I/O	ST	
RP9	22	28	33	I/O	ST	
RP10	31	39	49	I/O	ST	
RP11	46	58	72	I/O	ST	
RP12	45	57	71	I/O	ST	
RP13	14	18	23	I/O	ST	
RP14	29	35	43	I/O	ST	
RP15	—	43	53	I/O	ST	
RP16	33	41	51	I/O	ST	
RP17	32	40	50	I/O	ST	
RP18	11	15	20	I/O	ST	
RP19	6	8	12	I/O	ST	
RP20	53	67	82	I/O	ST	
RP21	4	6	10	I/O	ST	
RP22	51	63	78	I/O	ST	
RP23	50	62	77	I/O	ST	
RP24	49	61	76	I/O	ST	
RP25	52	66	81	I/O	ST	
RP26	5	7	11	I/O	ST	
RP27	8	10	14	I/O	ST	
RP28	12	16	21	I/O	ST	
RP29	30	36	44	I/O	ST	
RP30	—	42	52	I/O	ST	
RP31	—	—	39	I/O	ST	

图注： TTL = TTL 输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电平输入 / 输出　　I^2C™ = I^2C/SMBus 输入缓冲器

表 1-4： PIC24FJ256GA110 系列引脚配置说明 （续）

功能	引脚号			I/O	输入缓冲器	说明
	64 引脚 TQFP 封装	80 引脚 TQFP 封装	100 引脚 TQFP 封装			
RPI32	—	—	40	I	ST	可重映射的外设 (只能输入)。
RPI33	—	13	18	I	ST	
RPI34	—	14	19	I	ST	
RPI35	—	53	67	I	ST	
RPI36	—	52	66	I	ST	
RPI37	48	60	74	I	ST	
RPI38	—	4	6	I	ST	
RPI39	—	—	7	I	ST	
RPI40	—	5	8	I	ST	
RPI41	—	—	9	I	ST	
RPI42	—	64	79	I	ST	
RPI43	—	37	47	I	ST	
RPI44	—	44	54	I	ST	
RPI45	35	45	55	I	ST	
RTCC	42	54	68	O	—	实时时钟闹钟 / 秒脉冲输出。
SCL1	37	47	57	I/O	I^2C	I2C1 同步串行时钟输入 / 输出。
SCL2	32	52	58	I/O	I^2C	I2C2 同步串行时钟输入 / 输出。
SCL3	2	2	4	I/O	I^2C	I2C3 同步串行时钟输入 / 输出。
SDA1	36	46	56	I/O	I^2C	I2C1 数据输入 / 输出。
SDA2	31	53	59	I/O	I^2C	I2C2 数据输入 / 输出。
SDA3	3	3	5	I/O	I^2C	I2C3 数据输入 / 输出。
SOSCI	47	59	73	I	ANA	辅助振荡器 /Timer1 时钟输入。
SOSCO	48	60	74	O	ANA	辅助振荡器 /Timer1 时钟输出。
T1CK	48	60	74	I	ST	Timer1 时钟。
TCK	27	33	38	I	ST	JTAG 测试时钟 / 编程时钟输入。
TDI	28	34	60	I	ST	JTAG 测试数据 / 编程数据输入。
TDO	24	14	61	O	—	JTAG 测试数据输出。
TMS	23	13	17	I	ST	JTAG 测试模式选择输入。
VCAP	56	70	85	P	—	外部滤波电容连接 (使能稳压器)。
VDD	10,26,38	12,32,48	2,16,37, 46,62	P	—	外设数字逻辑和 I/O 引脚的正电源。
VDDCORE	56	70	85	P	—	单片机内核逻辑的正电源 (禁止稳压器)。
VREF-	15	23	28	I	ANA	A/D 和比较器参考电压 (低电平) 输入。
VREF+	16	24	29	I	ANA	A/D 和比较器参考电压 (高电平) 输入。
VSS	9, 25, 41	11, 31, 51	15, 36, 45, 65, 75	P	—	逻辑电路和 I/O 引脚的接地参考端。

图注： TTL = TTL 输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电平输入 / 输出　　I^2C™ = I^2C/SMBus 输入缓冲器

2.0 CPU

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的**“第 2 章 CPU”**（DS39703A_CN）。

PIC24F CPU 采用 16 位 （数据）的改进哈佛架构，具有增强的指令集和带有长度可变的操作码字段的24位指令字。程序计数器 （Program Counter， PC）为 23 位宽，可以寻址最大 4M 指令的用户程序存储空间。单周期指令预取机制可帮助维持吞吐量，并使指令的执行具有预测性。除了改变程序流的指令、双字传送（`MOV.D`）指令和表指令以外，所有指令都在单个周期内执行。使用 `REPEAT` 指令可以支持无开销的程序循环结构，在任何时候都可被中断。

PIC24F 器件在编程模型中有十六个 16 位的工作寄存器。每个工作寄存器都可以充当数据、地址或地址偏移量寄存器。第 16 个工作寄存器 （W15）用作软件堆栈指针 （Stack Pionter， SP），用于中断和调用。

数据存储空间的高 32 KB 可以映射到程序空间中，映射的边界可以是任何 16K 程序字边界，边界由 8 位的程序空间可视性页地址 （PSVPAG）寄存器定义。程序空间到数据空间的映射功能让任何指令都能象访问数据空间一样访问程序空间。

指令集架构 （Instruction Set Architecture， ISA）与 PIC18 的相比有了显著的提升，并保持了一定程度的向下兼容性。该架构直接或通过简单的宏支持所有的 PIC18 指令和寻址模式。对编译器执行效率的需求也促使了对 ISA 的许多改进。

内核支持固有 （无操作数）寻址、相对寻址、立即数寻址、存储器直接寻址及其他三组寻址模式。所有模式都支持寄存器直接和各种寄存器间接寻址模式。每组寻址模式都提供了最多七种寻址方式。指令根据其功能要求，与预定义的寻址模式相关。

对于大多数指令，能在每个指令周期内执行一次数据（或程序数据）存储器读操作、一次工作寄存器（数据）读操作、一次数据存储器写操作和一次程序（指令）存储器读操作。因此可以支持三个操作数的指令，使三个操作数的运算 （即， A + B = C）能在单周期内执行。

内核中包含一个高速的 17 位× 17 位乘法器可以极大地加强内核的运算能力和吞吐量。此乘法器支持有符号、无符号和混合模式的 16 位× 16 位或 8 位× 8 位整数乘法。所有的乘法指令都在单周期内执行。

已对 16 位 ALU 进行了改进使其具备一个支持整数除法的硬件，该硬件支持迭代的不可撤消的除法算法。它可以和 `REPEAT` 指令循环机制和迭代除法指令一起工作，支持 32 位 （或 16 位）除以 16 位有符号和无符号整数的除法运算。所有除法运算都需要 19 个周期来完成，但是可在任何周期边界被中断。

PIC24F 具有向量异常机制，带有最多 8 个不可屏蔽陷阱源和 118 个中断源。可以给每个中断源分配 7 个优先级之一。

CPU 的框图如图 2-1 所示。

2.1 编程模型

图 2-2 中所示为 PIC24F 的编程模型。编程模型中的所有寄存器都是存储器映射的，并且可以由指令直接操作。表 2-1 中提供了对每个寄存器的描述。所有与编程模型相关联的寄存器都是存储器映射的。

图 2-1： PIC24F CPU 内核框图

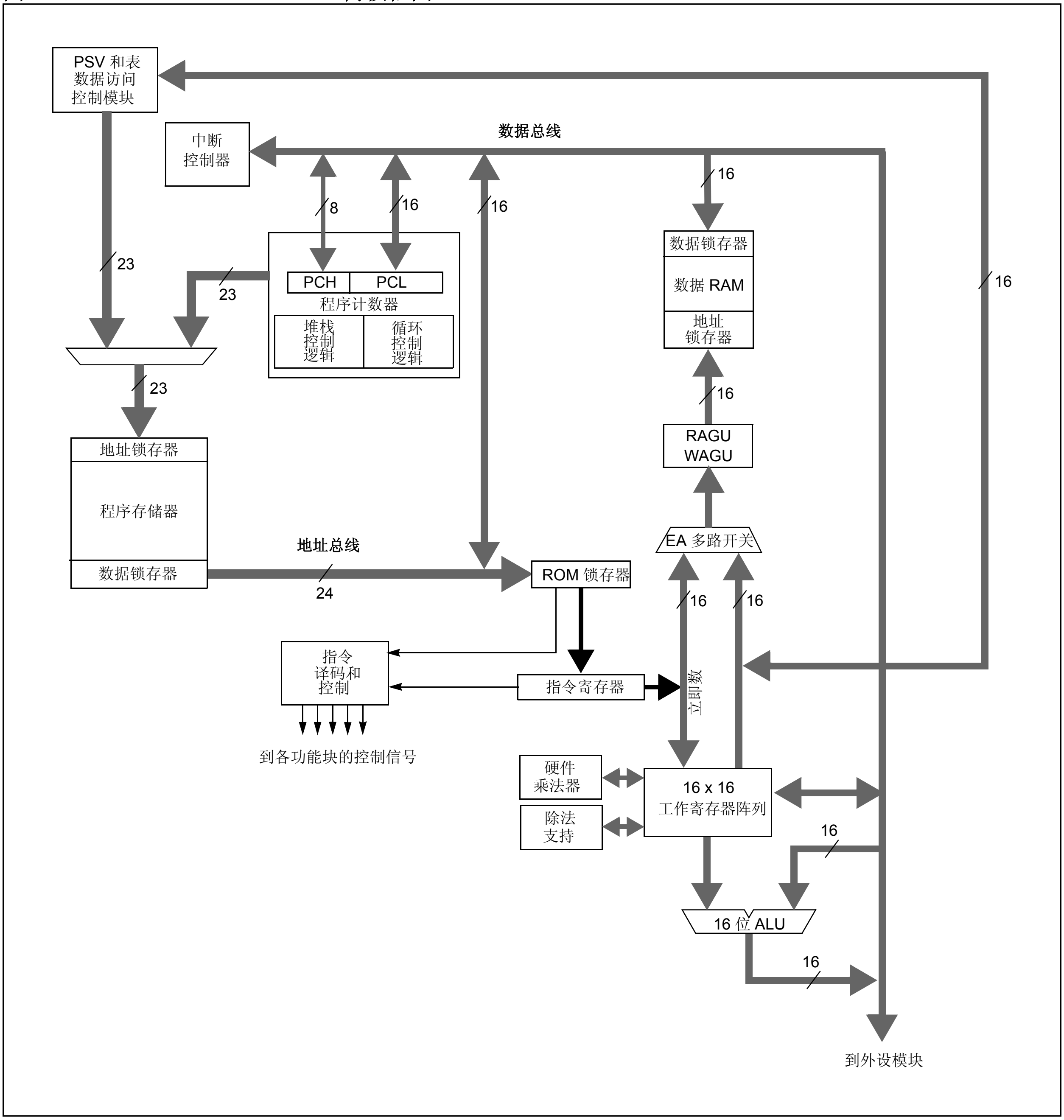
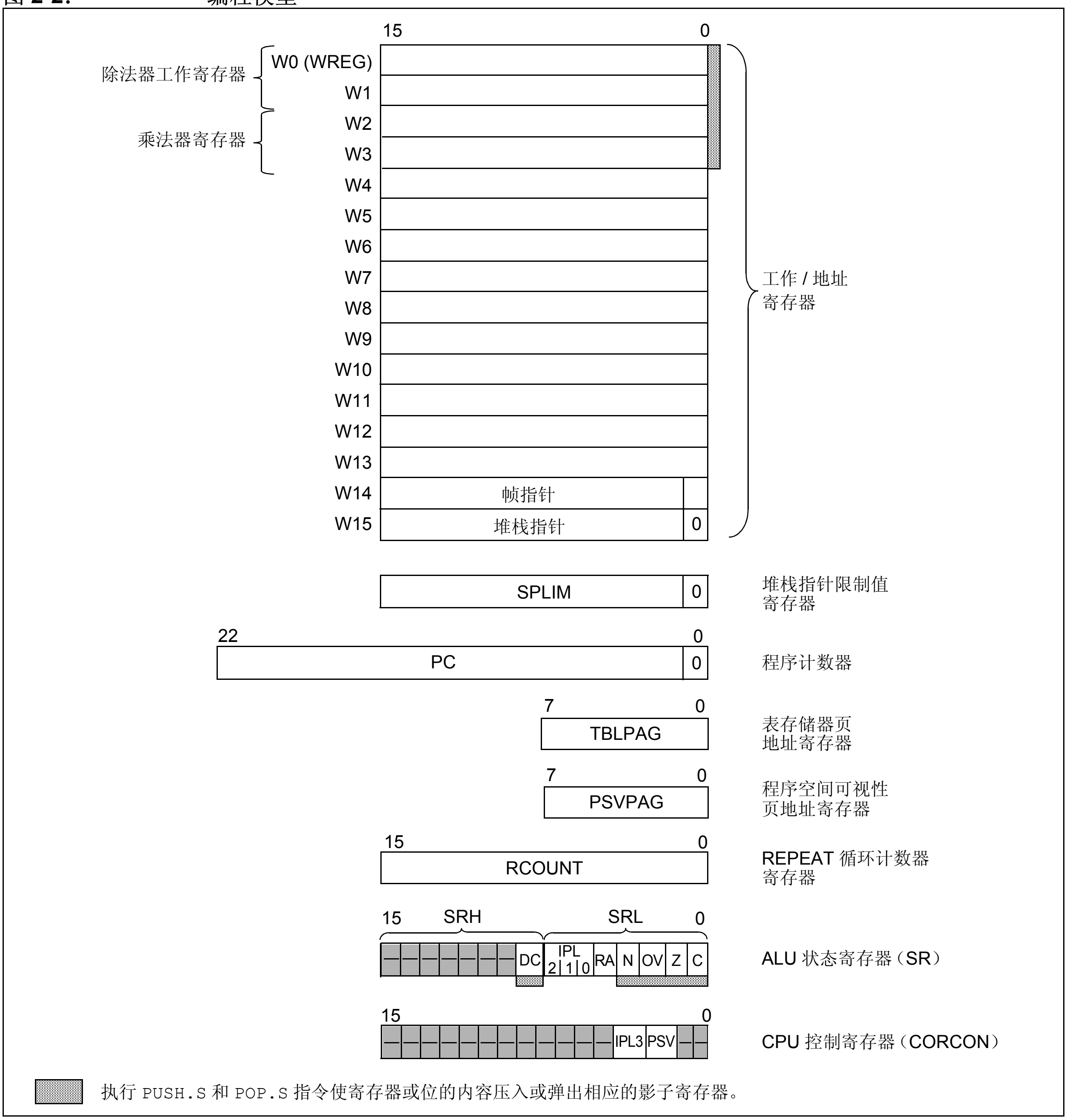


表 2-1： CPU 内核寄存器

寄存器名称	说明
W0 到 W15	工作寄存器阵列
PC	23 位程序计数器
SR	ALU 状态寄存器
SPLIM	堆栈指针限制寄存器
TBLPAG	表存储器页地址寄存器
PSVPAG	程序空间可视性页地址寄存器
RCOUNT	REPEAT 循环计数器寄存器
CORCON	CPU 控制寄存器

图 2-2： 编程模型



2.2 CPU 控制寄存器

寄存器 2-1： **SR：ALU 状态寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	DC
bit 15							bit 8

R/W-0(1)	R/W-0(1)	R/W-0(1)	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2(2)	IPL1(2)	IPL0(2)	RA	N	OV	Z	C
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-9 **未实现：**读为 0

bit 8 **DC：**ALU 半进位 / 借位标志位
1 = 结果的第 4 低位（对于字节大小的数据）或第 8 低位（对于字大小的数据）发生了向高位的进位
0 = 结果的第 4 低位或第 8 低位未发生向高位的进位

bit 7-5 **IPL2:IPL0：**CPU 中断优先级状态位 (1,2)
111 = CPU 中断优先级为 7（15）；禁止用户中断。
110 = CPU 中断优先级为 6（14）
101 = CPU 中断优先级为 5（13）
100 = CPU 中断优先级为 4（12）
011 = CPU 中断优先级为 3（11）
010 = CPU 中断优先级为 2（10）
001 = CPU 中断优先级为 1（9）
000 = CPU 中断优先级为 0（8）

bit 4 **RA：**REPEAT 循环状态位
1 = 正在进行 REPEAT 循环
0 = 不在进行 REPEAT 循环

bit 3 **N：**ALU 负标志位
1 = 结果为负
0 = 结果非负（零或正值）

bit 2 **OV：**ALU 溢出标志位
1 = 在本次算术运算中有符号（二进制补码）运算发生了溢出
0 = 未发生溢出

bit 1 **Z：**ALU 全零标志位
1 = 影响 Z 位的任何运算在过去某时已将该位置 1
0 = 影响 Z 位的最近一次运算已经将该位清零（即运算结果非零）

bit 0 **C：**ALU 进位 / $\overline{\text{借位}}$标志位
1 = 结果的最高位（Most Significant bit， MSb）发生了进位
0 = 结果的最高位未发生进位

注 **1：** 当 NSTDIS（INTCON1<15>）= 1 时 IPL 状态位为只读。
2： IPL 状态位与 IPL3 位（CORCON<3>）共同决定 CPU 的中断优先级（Interrupt Priority Level， IPL）。如果 IPL3 = 1，则括号中的值表示中断优先级。

寄存器 2-2： CORCON：CPU 控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0	R/W-0	U-0	U-0
—	—	—	—	IPL3[(1)]	PSV	—	—
bit 7							bit 0

图注：		U = 未实现位，读为 0	
R = 可读位	W = 可写位	C = 可清零位	
-n = 上电复位时的值	1 = 位置 1	0 = 清零	x = 未知

bit 15-4	**未实现：**读为 0
bit 3	**IPL3：**CPU 中断优先级状态位 [(1)] 1 = CPU 中断优先级大于 7 0 = CPU 中断优先级小于或等于 7
bit 2	**PSV：**数据空间的程序空间可视性使能位 1 = 程序空间在数据空间中可视 0 = 程序空间在数据空间中不可视
bit 1-0	**未实现：**读为 0

注 **1：** 当 IPL3 = 1 时，禁止用户中断。

2.3 算术逻辑单元 （ALU）

PIC24F ALU 为 16 位宽，并能进行加、减、移位和逻辑运算。除非另外指明，否则算术运算一般以 2 进制补码方式进行。根据不同的运算，ALU 可能会影响 SR 寄存器中的进位标志位（C）、全零标志位（Z）、负标志位（N）、溢出标志位（OV）和半进位标志位（DC）的值。在减法运算中，C 和 DC 位分别作为借位和半借位标志位。

根据所使用的指令模式，ALU 可以执行 8 位或 16 位运算。依据指令的寻址模式，ALU 运算的数据可以来自 W 寄存器阵列或数据存储器。同样，ALU 的输出数据可以被写入 W 寄存器阵列或数据存储单元。

PIC24F CPU 融入了对乘法和除法的硬件支持。它带有专用的硬件乘法器以及支持 16 位除数除法的硬件。

2.3.1 乘法器

ALU 包含一个高速的 17 位 x17 位乘法器。它支持以下各种无符号、有符号或混合符号乘法运算模式：

1. 16 位 x 16 位有符号乘法运算
2. 16 位 x 16 位无符号乘法运算
3. 16 位有符号数 x 5 位无符号立即数
4. 16 位无符号数 x 16 位无符号数
5. 16 位无符号数 x 5 位无符号立即数
6. 16 位无符号数 x 16 位有符号数
7. 8 位无符号数 x 8 位无符号数

2.3.2 除法器

除法模块支持具有下列数据长度的有符号整数和无符号整数的除法运算：

1. 32 位有符号数 /16 位有符号数
2. 32 位无符号数 /16 位无符号数
3. 16 位有符号数 /16 位有符号数
4. 16 位无符号数 /16 位无符号数

所有除法指令的商都被放在 W0 中，余数放在 W1 中。十六位有符号和无符号 DIV 指令可为 16 位除数指定任一 W 寄存器 （Wn），为 32 位被除数指定任意两个连续的 W 寄存器 （W(m+1):Wm）。除法运算中处理除数的每一位需要一个周期，因此 32 位 /16 位和 16 位 /16 位指令的执行周期数相同。

2.3.3 支持多位移位

PIC24F ALU 支持单位移位和单周期多位算术和逻辑移位操作。由一个移位寄存器模块执行多位移位，在单个周期内最多可将数据算术右移或左移 15 位。所有的多位移位指令仅支持源操作数和目标结果的寄存器直接寻址模式。

在下面的表 2-2 中汇总了所有使用移位操作的指令。

表 **2-2**： 使用单位和多位移位操作的指令

指令	说明
ASR	将源寄存器算术右移一位或多位。
SL	将源寄存器左移一位或多位。
LSR	将源寄存器逻辑右移一位或多位。

3.0 存储器构成

由于 PIC24F 单片机采用了哈佛架构，因此它具有独立的程序和数据存储空间以及独立的程序和数据总线。此架构还允许在代码执行的过程中直接通过数据空间访问程序存储空间。

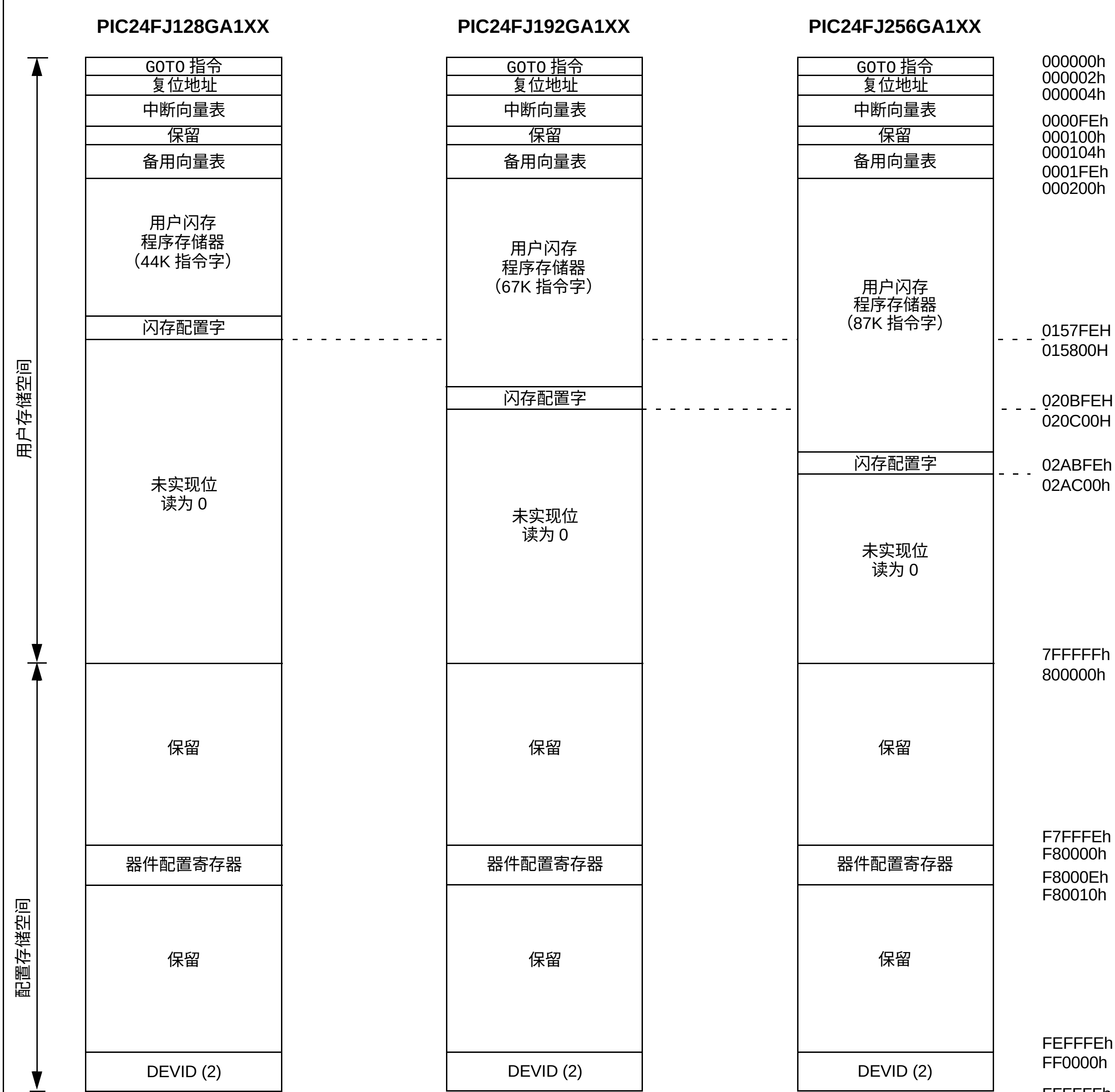
3.1 程序地址空间

PIC24FJ256GA110系列器件的程序地址空间可存储4M个指令字。可通过由程序执行过程中 23 位程序计数器（PC）、表操作或数据空间重新映射得到的 24 位值寻址这一空间，如**第 3.3 节“程序存储空间与数据存储空间的接口”**所示。

用户只能访问程序存储空间的低半地址部分（地址范围为 000000h 至 7FFFFFh）。使用 TBLRD/TBLWT 指令时，情况有所不同，这两条指令使用 TBLPAG<7> 位以允许访问配置存储空间中的配置位和器件 ID。

器件的 PIC24FJ256GA110 系列的存储器映射如图 3-1 所示。

图 3-1： PIC24FJ256GA110 系列器件的程序存储空间映射



注： 存储区未按比例显示。

3.1.1 程序存储器构成

程序存储空间由可字寻址的块构成。虽然它被视为 24 位宽，但将程序空间的每个地址视作一个低位字和一个高位字的组合更加合理，其中高位字的高字节部分未实现。低位字的地址始终为偶数，而高位字的地址为奇数 （见图 3-2）。

程序存储器地址始终在低位字处按字对齐。并且在代码执行过程中地址将递增或递减 2。这种寻址方式与数据空间寻址兼容，且为访问程序存储空间中的数据提供了可能。

3.1.2 硬存储器向量

所有 PIC24F 器件中从 00000h 到 000200h 之间的地址空间都是保留的，用来存储硬编码的程序执行向量。提供了一个硬件复位向量将代码执行从器件复位时 PC 的默认值重新定位到代码实际开始处。用户可在 000000h 地址编写一条 GOTO 指令以将代码的实际起始处定义为 000002h。

PIC24F 器件也具有 2 个中断向量表，地址范围分别为 000004h 至 0000FFh 和 000100h 至 0001FFh。这两个中断向量表允许使用不同的中断服务程序 （ISR）处理每个器件中断源。**第 6.1 节 " 中断向量表 "** 提供了有关中断向量表的更多详细信息。

3.1.3 闪存配置字

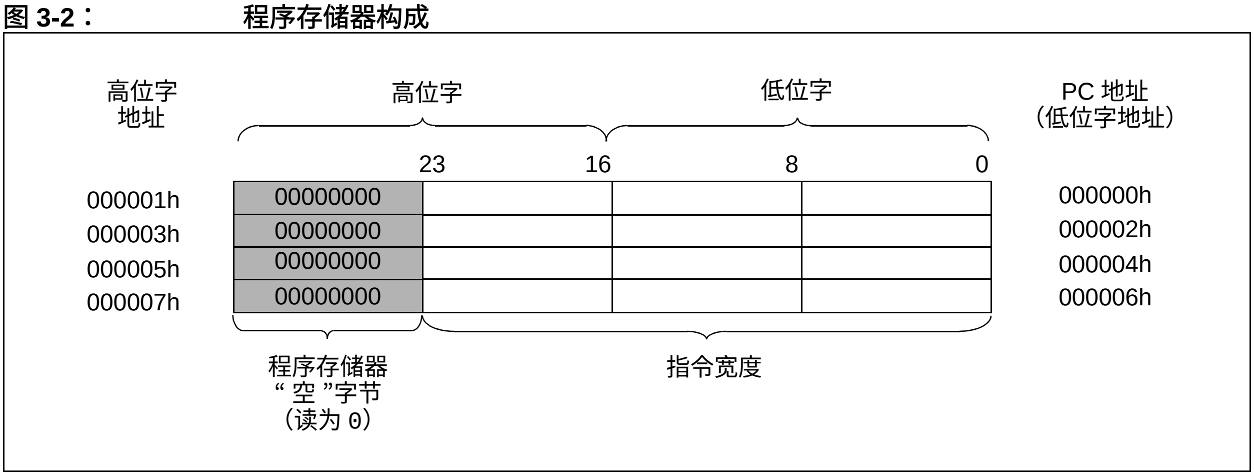
在 PIC24FJ256GA110 系列器件中，保留片上程序存储器的开始三个字用于配置信息。器件复位时，该配置信息被复制到相应的配置寄存器中。 PIC24FJ256GA110 系列中器件的闪存配置字的地址如表 3-1 所示。图中显示了它们在存储器映射图中的位置以及其他的存储器向量。

程序存储器中的配置字为紧凑的格式。实际配置位被映射到配置存储空间的几个不同的寄存器中。它们在闪存配置字中的顺序并不反映它们在配置空间中的相应顺序。**第 24.1 节 " 配置位 "** 给出了器件配置字的更多详细信息。

表 3-1： PIC24FJ256GA110 系列器件的闪存配置字

器件	程序存储器（字）	配置字地址
PIC24FJ128GA	44,032	0157FAh: 0157FEh
PIC24FJ192GA	67,072	020BFAh: 020BFEh
PIC24FJ256GA	87,552	02ABFAh: 02ABFEh

图 3-2： 程序存储器构成



 超前信息

3.2 数据地址空间

PIC24F的内核具有一个独立的16位宽的数据存储空间，可将其作为一个单独的线形范围寻址。使用两个地址发生单元（Address Generation Units，AGU）对数据空间进行读写操作。图 3-3 显示了数据空间的存储器映射。

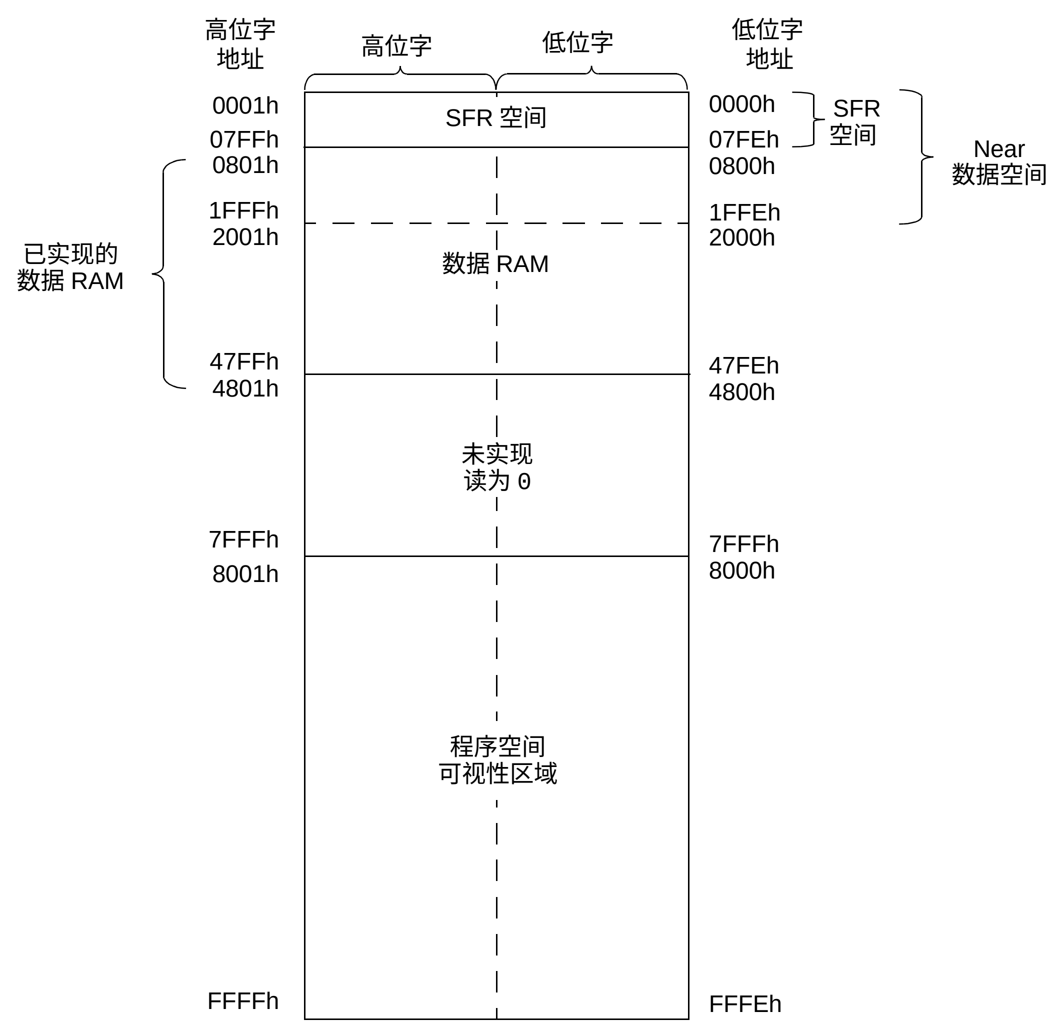
数据存储空间中的所有有效地址 （Effective Address，EA）都是 16 位宽，并且指向数据空间内的字节。这种构成方式使得数据空间的地址范围为 64 KB 或 32K 字。数据存储空间的低半部分 （即 EA<15> = 0）用作已实现的存储器地址单元，而高半部分（EA<15> = 1）则保留为程序空间可视性区域 （见**第 3.3.3 节 " 使用程序空间可视性来读程序存储器中的数据 "**）。

PIC24FJ256GA110 系列器件实现了总共 16 KB 的数据存储空间。如果 EA 指向此区域以外的存储单元，将返回全零的字或字节。

3.2.1 数据空间宽度

数据存储空间由可字节寻址的 16 位宽的块构成。在数据存储器和寄存器中的数据是以16位字为单位对齐的，但所有数据空间的有效地址（EA）都被解析为字节。每个字的低字节部分具有偶地址，而高字节部分则具有奇地址。

图 3-3： PIC24FJ256GA110 系列器件的数据空间存储器映射



注： 数据存储区未按比例显示。

3.2.2 数据存储器构成和对齐方式

为了保持与PIC® 器件向后兼容，并提高数据空间存储器的使用效率，PIC24F 指令集既支持字操作，也支持字节操作。字节访问会在内部对按字对齐的存储空间的所有有效地址（EA）计算进行调整。例如，对于执行后修改寄存器间接寻址模式 [Ws++]，字节操作时，内核将其识别为值 Ws + 1，而字操作时，内核将其识别为 Ws + 2。

使用任何有效地址的最低位 （LSb）来决定要选择哪个字节，数据字节的读操作将读取包含此字节的整个字。选中的字节将被放在数据路径的低字节处。也就是说，数据存储器和寄存器被组织为两个并行的共享（字）地址译码，而写入线相互独立的实体。数据字节写操作只写入存储阵列和寄存器中与字节地址匹配的相应部分。

所有字访问都必须按偶地址对齐。不支持不对齐的字数据读取操作。因此当进行混合字节和字操作或从 8 位 MCU 代码移植到此系列器件时，必须要小心。若试图进行不对齐的读或写操作，则会产生地址错误陷阱。如果在读操作时产生错误，正在执行的指令将完成；而如果在写操作时产生错误，指令仍将执行，但不会进行写入。无论是哪种情况都将执行陷阱，从而允许系统和 / 或用户检查地址错误发生之前的机器状态。

所有装入 W 寄存器的字节都将被装入到低字节（LSB）。高字节 （MSB）不变。

提供了一个符号扩展指令（SE），允许用户把 8 位的有符号数据转换为 16 位的有符号值。或者，对于 16 位无符号数据，用户可以清零任何 W 寄存器的 MSB，方法是在相应的地址处执行一条零扩展 （ZE）指令。

尽管大多数指令能够对字或字节数据进行操作，但必须要注意的是，某些指令仅适用于字操作。

3.2.3 NEAR 数据空间

将 0000h 和 1FFFh 之间的 8 KB 的区域称为 Near 数据空间。可以使用所有存储器直接寻址指令中的 13 位绝对地址直接寻址这一空间中的地址单元。剩余的数据空间通过间接寻址访问。此外，还可以使用 MOV 指令寻址整个数据空间，支持使用 16 位地址字段的存储器直接寻址。

3.2.4 SFR 空间

Near 数据空间的前 2 KB（0000h 至 07FFh）主要被特殊功能寄存器 （Special Function Register， SFR）占用。 PIC24F 的内核和外设模块使用这些寄存器来控制器件的操作。

SFR 分布在受其控制的大量模块中，通常一个模块会使用一组 SFR。许多 SFR 空间包含未用的地址单元，它们读为 0。 SFR 空间的框图显示了实际实现 SFR 的位置，如表 3-2 所示。每个实现的区域表示一个 32 字节区域，其中至少有一个地址实现为 SFR。表 3-3 至表 3-29 给出了所有已实现的 SFR 及其地址的完整列表。

表 3-2： SFR 数据空间的实现区域

	SFR 空间地址							
	xx00	**xx20**	**xx40**	**xx60**	**xx80**	**xxA0**	**xxC0**	**xxE0**
000h	内核			ICN	中断			—
100h	定时器		捕捉		比较			
200h	I^2C™	UART	SPI/UART	SPI/I^2C	SPI	UART	I/O	
300h	A/D	A/D/CTMU	—	—	—	—	—	—
400h	—	—	—	—	—			—
500h	—	—	—	—	—	—	—	—
600h	PMP	RTC/比较器	CRC	—	PPS			—
700h	—	—	系统	NVM/PMD	—	—	—	—

图注： — = 此存储块中未实现的 SFR。

 超前信息

表 3-3： CPU 内核寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
WREG0	0000	工作寄存器 0																0000
WREG1	0002	工作寄存器 1																0000
WREG2	0004	工作寄存器 2																0000
WREG3	0006	工作寄存器 3																0000
WREG4	0008	工作寄存器 4																0000
WREG5	000A	工作寄存器 5																0000
WREG6	000C	工作寄存器 6																0000
WREG7	000E	工作寄存器 7																0000
WREG8	0010	工作寄存器 8																0000
WREG9	0012	工作寄存器 9																0000
WREG10	0014	工作寄存器 10																0000
WREG11	0016	工作寄存器 11																0000
WREG12	0018	工作寄存器 12																0000
WREG13	001A	工作寄存器 13																0000
WREG14	001C	工作寄存器 14																0000
WREG15	001E	工作寄存器 15																0800
SPLIM	0020	堆栈指针限制值寄存器																xxxx
PCL	002E	程序计数器低位字寄存器																0000
PCH	0030	—	—	—	—	—	—	—	—	程序计数器寄存器的高字节								0000
TBLPAG	0032	—	—	—	—	—	—	—	—	表存储器页地址寄存器								0000
PSVPAG	0034	—	—	—	—	—	—	—	—	程序空间可视性页地址寄存器								0000
RCOUNT	0036	Repeat 循环计数器寄存器																xxxx
SR	0042	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	PSV	—	—	0000
DISICNT	0052	—	—	禁止中断计数器寄存器														xxxx

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-4： ICN 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
CNPD1	0054	CN15PDE	CN14PDE	CN13PDE	CN12PDE	CN11PDE	CN10PDE	CN9PDE	CN8PDE	CN7PDE	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	0056	CN31PDE	CN30PDE	CN29PDE	CN28PDE	CN27PDE	CN26PDE	CN25PDE	CN24PDE	CN23PDE	CN22PDE	CN21PDE[1]	CN20PDE[1]	CN19PDE[1]	CN18PDE	CN17PDE	CN16PDE	0000
CNPD3	0058	CN47PDE[1]	CN46PDE[2]	CN45PDE[1]	CN44PDE[1]	CN43PDE[1]	CN42PDE[1]	CN41PDE[1]	CN40PDE[2]	CN39PDE[2]	CN38PDE[2]	CN37PDE[2]	CN36PDE[2]	CN35PDE[2]	CN34PDE[2]	CN33PDE[2]	CN32PDE	0000
CNPD4	005A	CN63PDE	CN62PDE	CN61PDE	CN60PDE	CN59PDE	CN58PDE	CN57PDE[1]	CN56PDE	CN55PDE	CN54PDE	CN53PDE	CN52PDE	CN51PDE	CN50PDE	CN49PDE	CN48PDE[2]	0000
CNPD5	005C	CN79PDE[2]	CN78PDE[1]	CN77PDE[1]	CN76PDE[2]	CN75PDE[2]	CN74PDE[1]	CN73PDE[1]	CN72PDE	CN71PDE	CN70PDE[1]	CN69PDE	CN68PDE	CN67PDE[1]	CN66PDE[1]	CN65PDE	CN64PDE	0000
CNPD6	005E	—	—	—	—	—	—	—	—	—	—	—	CN84PDE	CN83PDE	CN82PDE[2]	CN81PDE[2]	CN80PDE[2]	0000
CNEN1	0060	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0062	CN31IE	CN30IE	CN29IE	CN28IE	CN27IE	CN26IE	CN25IE	CN24IE	CN23IE	CN22IE	CN21IE[1]	CN20IE[1]	CN19IE[1]	CN18IE	CN17IE	CN16IE	0000
CNEN3	0064	CN47IE[1]	CN46IE[2]	CN45IE[1]	CN44IE[1]	CN43IE[1]	CN42IE[1]	CN41IE[1]	CN40IE[2]	CN39IE[2]	CN38IE[2]	CN37IE[2]	CN36IE[2]	CN35IE[2]	CN34IE[2]	CN33IE[2]	CN32IE	0000
CNEN4	0066	CN63IE	CN62IE	CN61IE	CN60IE	CN59IE	CN58IE	CN57IE[1]	CN56IE	CN55IE	CN54IE	CN53IE	CN52IE	CN51IE	CN50IE	CN49IE	CN48IE[2]	0000
CNEN5	0068	CN79IE[2]	CN78IE[1]	CN77IE[1]	CN76IE[2]	CN75IE[2]	CN74IE[1]	CN73IE[1]	CN72IE	CN71IE	CN70IE[1]	CN69IE	CN68IE	CN67IE[1]	CN66IE[1]	CN65IE	CN64IE	0000
CNEN6	006A	—	—	—	—	—	—	—	—	—	—	—	CN84IE	CN83IE	CN82IE[2]	CN81IE[2]	CN80IE[2]	0000
CNPU1	006C	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	006E	CN31PUE	CN30PUE	CN29PUE	CN28PUE	CN27PUE	CN26PUE	CN25PUE	CN24PUE	CN23PUE	CN22PUE	CN21PUE[1]	CN20PUE[1]	CN19PUE[1]	CN18PUE	CN17PUE	CN16PUE	0000
CNPU3	0070	CN47PUE[1]	CN46PUE[2]	CN45PUE[1]	CN44PUE[1]	CN43PUE[1]	CN42PUE[1]	CN41PUE[1]	CN40PUE[2]	CN39PUE[2]	CN38PUE[2]	CN37PUE[2]	CN36PUE[2]	CN35PUE[2]	CN34PUE[2]	CN33PUE[2]	CN32PUE	0000
CNPU4	0072	CN63PUE	CN62PUE	CN61PUE	CN60PUE	CN59PUE	CN58PUE	CN57PUE[1]	CN56PUE	CN55PUE	CN54PUE	CN53PUE	CN52PUE	CN51PUE	CN50PUE	CN49PUE	CN48PUE[2]	0000
CNPU5	0074	CN79PUE[2]	CN78PUE[1]	CN77PUE[1]	CN76PUE[2]	CN75PUE[2]	CN74PUE[1]	CN73PUE[1]	CN72PUE	CN71PUE	CN70PUE[1]	CN69PUE	CN68PUE	CN67PUE[1]	CN66PUE[1]	CN65PUE	CN64PUE	0000
CNPU6	0076	—	—	—	—	—	—	—	—	—	—	—	CN84PUE	CN83PUE	CN82PUE[2]	CN81PUE[2]	CN80PUE[2]	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

注 1： 在 64 引脚器件上未实现；读为 0。

2： 在 64 引脚和 80 引脚器件上未实现；读为 0。

表 3-5：　　中断控制器的寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
INTCON1	0080	NSTDIS	—	—	—	—	—	—	—	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—	IC8IF	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF	0000
IFS2	0088	—	—	PMPIF	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF	IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF	0000
IFS3	008A	—	RTCIF	—	—	—	—	—	—	—	INT4IF	INT3IF	—	—	MI2C2IF	SI2C2IF	—	0000
IFS4	008C	—	—	CTMUIF	—	—	—	—	LVDIF	—	—	—	—	CRCIF	U2ERIF	U1ERIF	—	0000
IFS5	008E	—	—	IC9IF	OC9IF	SPI3IF	SPF3IF	U4TXIF	U4RXIF	U4ERIF	—	MI2C3IF	SI2C3IF	U3TXIF	U3RXIF	U3ERIF	—	0000
IEC0	0094	—	—	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	—	IC8IE	IC7IE	—	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE	0000
IEC2	0098	—	—	PMPIE	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE	IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE	0000
IEC3	009A	—	RTCIE	—	—	—	—	—	—	—	INT4IE	INT3IE	—	—	MI2C2IE	SI2C2IE	—	0000
IEC4	009C	—	—	CTMUIE	—	—	—	—	LVDIE	—	—	—	—	CRCIE	U2ERIE	U1ERIE	—	0000
IEC5	009E	—	—	IC9IE	OC9IE	SPI3IE	SPF3IE	U4TXIE	U4RXIE	U4ERIE	—	MI2C3IE	SI2C3IE	U3TXIE	U3RXIE	U3ERIE	—	0000
IPC0	00A4	—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0	—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	00A6	—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0	—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—	4440
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0	4444
IPC3	00AA	—	—	—	—	—	—	—	—	—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	0044
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0	—	MI2C1P2	MI2C1P1	MI2C1P0	—	SI2C1P2	SI2C1P1	SI2C1P0	4444
IPC5	00AE	—	IC8IP2	IC8IP1	IC8IP0	—	IC7IP2	IC7IP1	IC7IP0	—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0	4404
IPC6	00B0	—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0	—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—	4440
IPC7	00B2	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0	4444
IPC8	00B4	—	—	—	—	—	—	—	—	—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0	0044
IPC9	00B6	—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0	—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—	4440
IPC10	00B8	—	OC7IP2	OC7IP1	OC7IP0	—	OC6IP2	OC6IP1	OC6IP0	—	OC5IP2	OC5IP1	OC5IP0	—	IC6IP2	IC6IP1	IC6IP0	4444
IPC11	00BA	—	—	—	—	—	—	—	—	—	PMPIP2	PMPIP1	PMPIP0	—	OC8IP2	OC8IP1	OC8IP0	0044
IPC12	00BC	—	—	—	—	—	MI2C2P2	MI2C2P1	MI2C2P0	—	SI2C2P2	SI2C2P1	SI2C2P0	—	—	—	—	0440
IPC13	00BE	—	—	—	—	—	INT4IP2	INT4IP1	INT4IP0	—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—	0440
IPC15	00C2	—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0	—	—	—	—	—	—	—	—	0400
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0	0004
IPC19	00CA	—	—	—	—	—	—	—	—	—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—	0040
IPC20	00CC	—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0	—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—	4440
IPC21	00CE	—	U4ERIP2	U4ERIP1	U4ERIP0	—	—	—	—	—	MI2C3P2	MI2C3P1	MI2C3P0	—	SI2C3P2	SI2C3P1	SI2C3P0	4044
IPC22	00D0	—	SPI3IP2	SPI3IP1	SPI3IP0	—	SPF3IP2	SPF3IP1	SPF3IP0	—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0	4444
IPC23	00D2	—	—	—	—	—	—	—	—	—	IC9IP2	IC9IP1	IC9IP0	—	OC9IP2	OC9IP1	OC9IP0	0044

图注：　　— = 未实现，读为 0。复位值以十六进制形式显示。

超前信息

表 3-6： 定时器寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
TMR1	0100	Timer1 寄存器																0000
PR1	0102	Timer1 周期寄存器																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	Timer2 寄存器																0000
TMR3HLD	0108	Timer3 保持寄存器 （仅在 32 位定时器工作模式下使用）																0000
TMR3	010A	Timer3 寄存器																0000
PR2	010C	Timer2 周期寄存器																FFFF
PR3	010E	Timer3 周期寄存器																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	0114	Timer4 寄存器																0000
TMR5HLD	0116	Timer5 保持寄存器 （仅在 32 位定时器工作模式下使用）																0000
TMR5	0118	Timer5 寄存器																0000
PR4	011A	Timer4 周期寄存器																FFFF
PR5	011C	Timer5 周期寄存器																FFFF
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-7： 输入捕捉寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
IC1CON1	0140	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC1CON2	0142	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
IC2BUF	014C	输入捕捉 1 缓冲寄存器																0000
IC1TMR	0146	定时器值 1 寄存器																xxxx
IC2CON1	0148	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC2CON2	014A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
IC2BUF	014C	输入捕捉 2 缓冲寄存器																0000
IC2TMR	014E	定时器值 2 寄存器																xxxx
IC3CON1	0150	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC3CON2	0152	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
IC3BUF	0154	输入捕捉 3 缓冲寄存器																0000
IC3TMR	0156	定时器值 3 寄存器																xxxx
IC4CON1	0158	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC4CON2	015A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
IC4BUF	015C	输入捕捉 4 缓冲寄存器																0000
IC4TMR	015E	定时器值 4 寄存器																xxxx
IC5CON1	0160	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC5CON2	0162	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
IC5BUF	0164	输入捕捉 5 缓冲寄存器																0000
IC5TMR	0166	定时器值 5 寄存器																xxxx
IC6CON1	0168	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC6CON2	016A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
IC6BUF	016C	输入捕捉 6 缓冲寄存器																0000
IC6TMR	016E	定时器值 6 寄存器																xxxx
IC7CON1	0170	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC7CON2	0172	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
IC7BUF	0174	输入捕捉 7 缓冲寄存器																0000
IC7TMR	0176	定时器值 7 寄存器																xxxx
IC8CON1	0178	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC8CON2	017A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
IC8BUF	017C	输入捕捉 8 缓冲寄存器																0000
IC8TMR	017E	定时器值 8 寄存器																xxxx
IC9CON1	0180	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC9CON2	0182	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
IC9BUF	0184	输入捕捉 9 缓冲寄存器																0000
IC9TMR	0186	定时器值 9 寄存器																xxxx

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-8： 输出比较寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
OC1CON1	0190	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC1CON2	0192	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
OC1RS	0194	输出比较 1 辅助寄存器																0000
OC1R	0196	输出比较 1 寄存器																0000
OC1TMR	0198	定时器值 1 寄存器																xxxx
OC2CON1	019A	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC2CON2	019C	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
OC2RS	019E	输出比较 2 辅助寄存器																0000
OC2R	01A0	输出比较 2 寄存器																0000
OC2TMR	01A2	定时器值 2 寄存器																xxxx
OC3CON1	01A4	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC3CON2	01A6	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
OC3RS	01A8	输出比较 3 辅助寄存器																0000
OC3R	01AA	输出比较 3 寄存器																0000
OC3TMR	01AC	定时器值 3 寄存器																xxxx
OC4CON1	01AE	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC4CON2	01B0	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
OC4RS	01B2	输出比较 4 辅助寄存器																0000
OC4R	01B4	输出比较 4 寄存器																0000
OC4TMR	01B6	定时器值 4 寄存器																xxxx
OC5CON1	01B8	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC5CON2	01BA	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
OC5RS	01BC	输出比较 5 辅助寄存器																0000
OC5R	01BE	输出比较 5 寄存器																0000
OC5TMR	01C0	定时器值 5 寄存器																xxxx
OC6CON1	01C2	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC6CON2	01C4	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
OC6RS	01C6	输出比较 6 辅助寄存器																0000
OC6R	01C8	输出比较 6 寄存器																0000
OC6TMR	01CA	定时器值 6 寄存器																xxxx
OC7CON1	01CC	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC7CON2	01CE	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
OC7RS	01D0	输出比较 7 辅助寄存器																0000
OC7R	01D2	输出比较 7 寄存器																0000
OC7TMR	01D4	定时器值 7 寄存器																xxxx

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-8： 输出比较寄存器映射 （续）

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
OC8CON1	01D6	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC8CON2	01D8	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
OC8RS	01DA	输出比较 8 辅助寄存器																0000
OC8R	01DC	输出比较 8 寄存器																0000
OC8TMR	01DE	定时器值 8 寄存器																xxxx
OC9CON1	01E0	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC9CON2	01E2	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
OC9RS	01E4	输出比较 9 辅助寄存器																0000
OC9R	01E6	输出比较 9 寄存器																0000
OC9TMR	01E8	定时器值 9 寄存器																xxxx

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-9： I²C 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
I2C1RCV	0200	—	—	—	—	—	—	—	—	接收寄存器								0000
I2C1TRN	0202	—	—	—	—	—	—	—	—	发送寄存器								00FF
I2C1BRG	0204	—	—	—	—	—	—	—	波特率发生器寄存器									0000
I2C1CON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1STAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	$D/\overline{A}$	P	S	$R/\overline{W}$	RBF	TBF	0000
I2C1ADD	020A	—	—	—	—	—	—	地址寄存器										0000
I2C1MSK	020C	—	—	—	—	—	—	地址掩码寄存器										0000
I2C2RCV	0210	—	—	—	—	—	—	—	—	接收寄存器								0000
I2C2TRN	0212	—	—	—	—	—	—	—	—	发送寄存器								00FF
I2C2BRG	0214	—	—	—	—	—	—	—	波特率发生器寄存器									0000
I2C2CON	0216	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C2STAT	0218	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	$D/\overline{A}$	P	S	$R/\overline{W}$	RBF	TBF	0000
I2C2ADD	021A	—	—	—	—	—	—	地址寄存器										0000
I2C2MSK	021C	—	—	—	—	—	—	地址掩码寄存器										0000
I2C3RCV	0270	—	—	—	—	—	—	—	—	接收寄存器								0000
I2C3TRN	0272	—	—	—	—	—	—	—	—	发送寄存器								00FF
I2C3BRG	0274	—	—	—	—	—	—	—	波特率发生器寄存器									0000
I2C3CON	0276	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C3STAT	0278	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	$D/\overline{A}$	P	S	$R/\overline{W}$	RBF	TBF	0000
I2C3ADD	027A	—	—	—	—	—	—	地址寄存器										0000
I2C3MSK	027C	—	—	—	—	—	—	地址掩码寄存器										0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-10： UART 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U1TXREG	0224	—	—	—	—	—	—	—	发送寄存器									xxxx
U1RXREG	0226	—	—	—	—	—	—	—	接收寄存器									0000
U1BRG	0228	波特率发生器预分频器																0000
U2MODE	0230	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U2TXREG	0234	—	—	—	—	—	—	—	发送寄存器									xxxx
U2RXREG	0236	—	—	—	—	—	—	—	接收寄存器									0000
U2BRG	0238	波特率发生器预分频器																0000
U3MODE	0250	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U3STA	0252	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U3TXREG	0254	—	—	—	—	—	—	—	发送寄存器									xxxx
U3RXREG	0256	—	—	—	—	—	—	—	接收寄存器									0000
U3BRG	0258	波特率发生器预分频器																0000
U4MODE	02B0	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U4STA	02B2	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U4TXREG	02B4	—	—	—	—	—	—	—	发送寄存器									xxxx
U4RXREG	02B6	—	—	—	—	—	—	—	接收寄存器									0000
U4BRG	02B8	波特率发生器预分频器																0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-11： SPI 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
SPI1STAT	0240	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI1CON1	0242	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI1CON2	0244	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI1BUF	0248	发送和接收缓冲器																0000
SPI2STAT	0260	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI2CON1	0262	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI2CON2	0264	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI2BUF	0268	发送和接收缓冲器																0000
SPI3STAT	0280	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI3CON1	0282	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI3CON2	0284	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI3BUF	0288	发送和接收缓冲器																0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-12： PORTA 寄存器映射 (1)

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7(2)	Bit 6(2)	Bit 5(2)	Bit 4(2)	Bit 3(2)	Bit2(2)	Bit 1(2)	Bit 0(2)	所有复位下的状态
TRISA	02C0	TRISA15	TRISA14	—	—	—	TRISA10	TRISA9	—	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	36FF
PORTA	02C2	RA15	RA14	—	—	—	RA10	RA9	—	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	LATA15	LATA14	—	—	—	LATA10	LATA9	—	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	02C6	ODA15	ODA14	—	—	—	ODA10	ODA9	—	ODA7	ODA6	ODA5	ODA4	ODA3	ODA2	ODA1	ODA0	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。显示的复位值针对 100 引脚器件。

注 1： PORTA 和所有相关位在 64 引脚器件上未实现，读为 0。除非另外声明，否则这些位仅在 80 引脚和 100 引脚器件上可用。

2： 这些位仅在 100 引脚器件上实现，否则读为 0。

表 3-13： PORTB 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx
ODCB	02CE	ODB15	ODB14	ODB13	ODB12	ODB11	ODB10	ODB9	ODB8	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0	0000

图注： 复位值以十六进制形式显示。

表 3-14： PORTC 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4(1)	Bit 3(2)	Bit 2(1)	Bit 1(2)	Bit 0	所有复位下的状态
TRISC	02D0	TRISC15	TRISC14	TRISC13	TRISC12	—	—	—	—	—	—	—	TRISC4	TRISC3	TRISC2	TRISC1	—	F01E
PORTC	02D2	RC15(3,4)	RC14	RC13	RC12(3)	—	—	—	—	—	—	—	RC4	RC3	RC2	RC1	—	xxxx
LATC	02D4	LATC15	LATC14	LATC13	LATC12	—	—	—	—	—	—	—	LATC4	LATC3	LATC2	LATC1	—	xxxx
ODCC	02D6	ODC15	ODC14	ODC13	ODC12	—	—	—	—	—	—	—	ODC4	ODC3	ODC2	ODC1	—	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。显示的复位值针对 100 引脚器件。

注 1： 在 64 引脚和 80 引脚器件上未实现；读为 0。

2： 在 64 引脚器件上未实现；读为 0。

3： RC12 和 RC15 仅在主振荡器被禁用或者当选择 EC 模式时 （POSCMD1:POSCMD0 配置位 = 11 或 00）才可用；否则读为 0。

4： RC15 仅在 POSCMD1:POSCMD0 配置位 = 11 或 00 且 OSCIOFN 配置位 = 1 时才可用。

表 3-15： PORTD 寄存器映射

文件名称	地址	Bit 15(1)	Bit 14(1)	Bit 13(1)	Bit 12(1)	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
TRISD	02D8	TRISD15	TRISD14	TRISD13	TRISD12	TRISD11	TRISD10	TRISD9	TRISD8	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	FFFF
PORTD	02DA	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx
LATD	02DC	LATD15	LATD14	LATD13	LATD12	LATD11	LATD10	LATD9	LATD8	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx
ODCD	02DE	ODD15	ODD14	ODD13	ODD12	ODD11	ODD10	ODD9	ODD8	ODD7	ODD6	ODD5	ODD4	ODD3	ODD2	ODD1	ODD0	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。显示的复位值针对 100 引脚器件。

注 1： 在 64 引脚的器件上未实现；读为 0。

表 3-16： PORTE 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9[(1)]	Bit 8[(1)]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
TRISE	02E0	—	—	—	—	—	—	TRISE9	TRISE8	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	03FF
PORTE	02E2	—	—	—	—	—	—	RE9	RE8	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	xxxx
LATE	02E4	—	—	—	—	—	—	LATE9	LATE8	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx
ODCE	02E6	—	—	—	—	—	—	ODE9	ODE8	ODE7	ODE6	ODE5	ODE4	ODE3	ODE2	ODE1	ODE0	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。显示的复位值针对 100 引脚器件。
注 1 ：在 64 引脚的器件上未实现；读为 0。

表 3-17： PORTF 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13[(1)]	Bit 12[(1)]	Bit 11	Bit 10	Bit 9	Bit 8[(2)]	Bit 7[(2)]	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
TRISF	02E8	—	—	TRISF13	TRISF12	—	—	—	TRISF8	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	31FF
PORTF	02EA	—	—	RF13	RF12	—	—	—	RF8	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	xxxx
LATF	02EC	—	—	LATF13	LATF12	—	—	—	LATF8	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	xxxx
ODCF	02EE	—	—	ODF13	ODF12	—	—	—	ODF8	ODF7	ODF6	ODF5	ODF4	ODF3	ODF2	ODF1	ODF0	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。显示的复位值针对 100 引脚器件。
注 1： 在 64 引脚和 80 引脚器件上未实现；读为 0。
2： 在 64 引脚器件上未实现；读为 0。

表 3-18： PORTG 寄存器映射

文件名称	地址	Bit 15[(1)]	Bit 14[(1)]	Bit 13[(1)]	Bit 12[(1)]	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1[(1)]	Bit 0[(1)]	所有复位下的状态
TRISG	02F0	TRISG15	TRISG14	TRISG13	TRISG12	—	—	TRISG9	TRISG8	TRISG7	TRISG6	—	—	TRISG3	TRISG2	TRISG1	TRISG0	F3CF
PORTG	02F2	RG15	RG14	RG13	RG12	—	—	RG9	RG8	RG7	RG6	—	—	RG3	RG2	RG1	RG0	xxxx
LATG	02F4	LATG15	LATG14	LATG13	LATG12	—	—	LATG9	LATG8	LATG7	LATG6	—	—	LATG3	LATG2	LATG1	LATG0	xxxx
ODCG	02F6	ODG15	ODG14	ODG13	ODG12	—	—	ODG9	ODG8	ODG7	ODG6	—	—	ODG3	ODG2	ODG1	ODG0	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。显示的复位值针对 100 引脚器件。
注 1： 在 64 引脚和 80 引脚器件上未实现；读为 0。

表 3-19： 填充配置寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
PADCFG1	02FC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTSECSEL	PMPTTL	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-20： ADC 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
ADC1BUF0	0300	ADC 数据缓冲器 0																xxxx
ADC1BUF1	0302	ADC 数据缓冲器 1																xxxx
ADC1BUF2	0304	ADC 数据缓冲器 2																xxxx
ADC1BUF3	0306	ADC 数据缓冲器 3																xxxx
ADC1BUF4	0308	ADC 数据缓冲器 4																xxxx
ADC1BUF5	030A	ADC 数据缓冲器 5																xxxx
ADC1BUF6	030C	ADC 数据缓冲器 6																xxxx
ADC1BUF7	030E	ADC 数据缓冲器 7																xxxx
ADC1BUF8	0310	ADC 数据缓冲器 8																xxxx
ADC1BUF9	0312	ADC 数据缓冲器 9																xxxx
ADC1BUFA	0314	ADC 数据缓冲器 10																xxxx
ADC1BUFB	0316	ADC 数据缓冲器 11																xxxx
ADC1BUFC	0318	ADC 数据缓冲器 12																xxxx
ADC1BUFD	031A	ADC 数据缓冲器 13																xxxx
ADC1BUFE	031C	ADC 数据缓冲器 14																xxxx
ADC1BUFF	031E	ADC 数据缓冲器 15																xxxx
AD1CON1	0320	ADON	—	ADSIDL	—	—	—	FORM1	FORM0	SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—	BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0324	ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS0	0328	CH0NB	—	—	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1PCFG	032C	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	0330	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000
AD1CSSH	0332	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CSS17	CSS16	0000

图注： — = 未实现，读为 0， r = 保留，保持为 0。复位值以十六进制形式显示。

表 3-21： CTMU 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
CTMUCON	033C	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000
CTMUICON	033E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

超前信息

表 3-22： 并行主 / 从端口寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
PMCON	0600	PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN	CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP	0000
PMMODE	0602	BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0	0000
PMADDR	0604	CS2	CS1	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	0000
PMDOUT1		并行端口数据输出寄存器 1 （缓冲器 0 和 1）																0000
PMDOUT2	0606	并行端口数据输出寄存器 2 （缓冲器 2 和 3）																0000
PMDIN1	0608	并行端口数据输入寄存器 1 （缓冲器 0 和 1）																0000
PMDIN2	060A	并行端口数据输入寄存器 2 （缓冲器 2 和 3）																0000
PMAEN	060C	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	0000
PMSTAT	060E	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-23： 实时时钟和日历寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
ALRMVAL	0620	基于 ALRMPTR<1:0> 的闹钟值寄存器窗口																xxxx
ALCFGRPT	0622	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVAL	0624	基于 RTCPTR<1:0> 的 RTCC 值寄存器窗口																xxxx
RCFGCAL	0626	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-24： 比较器寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
CMSTAT	0630	CMIDL	—	—	—		C3EVT	C2EVT	C1EVT	—	—	—	—	—	C3OUT	C2OUT	C1OUT	0000
CVRCON	0632	—	—	—	—	—	—	—	—	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000
CM1CON	0634	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000
CM2CON	0636	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000
CM3CON	0638	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-25： CRC 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
CRCCON	0640	—	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0	CRCFUL	CRCMPT	—	CRCGO	PLEN3	PLEN2	PLEN1	PLEN0	0040
CRCXOR	0642	X15	X14	X13	X12	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	—	0000
CRCDAT	0644	CRC 数据输入寄存器																0000
CRCWDAT	0646	CRC 结果寄存器																0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

表 3-26： 外设引脚选择寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
RPINR0	0680	—	—	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0	—	—	—	—	—	—	—	—	3F00
RPINR1	0682	—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0	—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0	3F3F
RPINR2	0684	—	—	T1CKR5	T1CKR4	T1CKR3	T1CKR2	T1CKR1	T1CKR0	—	—	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0	3F3F
RPINR3	0686	—	—	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0	—	—	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0	3F3F
RPINR4	0688	—	—	T5CKR5	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0	—	—	T4CKR5	T4CKR4	T4CKR3	T4CKR2	T4CKR1	T4CKR0	3F3F
RPINR7	068E	—	—	IC2R5	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0	—	—	IC1R5	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0	3F3F
RPINR8	0690	—	—	IC4R5	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0	—	—	IC3R5	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0	3F3F
RPINR9	0692	—	—	IC6R5	IC6R4	IC6R3	IC6R2	IC6R1	IC6R0	—	—	IC5R5	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0	3F3F
RPINR10	0694	—	—	IC8R5	IC8R4	IC8R3	IC8R2	IC8R1	IC8R0	—	—	IC7R5	IC7R4	IC7R3	IC7R2	IC7R1	IC7R0	3F3F
RPINR11	0696	—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0	—	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0	3F3F
RPINR15	069E	—	—	IC9R5	IC9R4	IC9R3	IC9R2	IC9R1	IC9R0	—	—	—	—	—	—	—	—	3F00
RPINR17	06A2	—	—	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0	—	—	—	—	—	—	—	—	3F00
RPINR18	06A4	—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0	—	—	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0	3F3F
RPINR19	06A6	—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0	—	—	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0	3F3F
RPINR20	06A8	—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0	—	—	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0	3F3F
RPINR21	06AA	—	—	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0	—	—	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0	3F3F
RPINR22	06AC	—	—	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0	—	—	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0	3F3F
RPINR23	06AE	—	—	—	—	—	—	—	—	—	—	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0	3F3F
RPINR27	06B6	—	—	U4CTSR5	U4CTSR4	U4CTSR3	U4CTSR2	U4CTSR1	U4CTSR0	—	—	U4RXR5	U4RXR4	U4RXR3	U4RXR2	U4RXR1	U4RXR0	3F3F
RPINR28	06B8	—	—	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0	—	—	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0	003F
RPINR29	06BA	—	—	—	—	—	—	—	—	—	—	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0	003F
RPOR0	06C0	—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0	—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0	0000
RPOR1	06C2	—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0	—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0	0000
RPOR2	06C4	—	—	RP5R5(1)	RP5R4(1)	RP5R3(1)	RP5R2(1)	RP5R1(1)	RP5R0(1)	—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0	0000
RPOR3	06C6	—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0	—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0	0000
RPOR4	06C8	—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0	—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0	0000
RPOR5	06CA	—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0	—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0	0000
RPOR6	06CC	—	—	RP13R5	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0	—	—	RP12R5	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0	0000
RPOR7	06CE	—	—	RP15R5(1)	RP15R4(1)	RP15R3(1)	RP15R2(1)	RP15R1(1)	RP15R0(1)	—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0	0000
RPOR8	06D0	—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0	—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0	0000
RPOR9	06D2	—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0	—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0	0000
RPOR10	06D4	—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0	—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0	0000
RPOR11	06D6	—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0	—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0	0000
RPOR12	06D8	—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0	—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0	0000
RPOR13	06DA	—	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0	—	—	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0	0000
RPOR14	06DC	—	—	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0	—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0	0000
RPOR15	06DE	—	—	RP31R5(2)	RP31R4(2)	RP31R3(2)	RP31R2(2)	RP31R1(2)	RP31R0(2)	—	—	RP30R5	RP30R4	RP30R3	RP30R2	RP30R1	RP30R0	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

注 1： 在 64 引脚器件上未实现；读为 0。

2： 在 64 引脚和 80 引脚器件上未实现；读为 0。

表 3-27： 系统寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
RCON	0740	TRAPR	IOPUWR	—	—	—	—	CM	VREGS	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	注 1
OSCCON	0742	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	IOLOCK	LOCK	—	CF	POSCEN	SOSCEN	OSWEN	注 2
CLKDIV	0744	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	—	—	—	—	—	—	—	—	0100
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000
REFOCON	074E	ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	—	—	—	—	—	—	—	—	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。
注 1： RCON 寄存器的复位值取决于复位事件的类型。具体信息，请参见**第 5.0 节 “ 复位 ”**。
2： OSCCON 寄存器的复位值取决于复位事件的类型和器件配置。具体信息，请参见**第 7.0 节 “ 振荡器配置 ”**。

表 3-28： NVM 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	—	—	ERASE	—	—	NVMOP3	NVMOP2	NVMOP1	NVMOP0	0000[1]
NVMKEY	0766	—	—	—	—	—	—	—	—	NVMKEY<7:0>								0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。
注 1： 显示的复位值仅用于 POR。在其他复位状态下的值取决于复位时对存储器执行写或擦除操作的状态。

表 3-29： PMD 寄存器映射

文件名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位下的状态
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	—	—	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD	0000
PMD2	0772	IC8MD	IC7MD	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	OC8MD	OC7MD	OC6MD	OC5MD	OC4MD	OC3MD	OC2MD	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	RTCCMD	PMPMD	CRCMD	—	—	—	U3MD	I2C3MD	I2C2MD	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	—	—	U4MD	—	REFOMD	CTMUMD	LVDMD	—	0000
PMD5	0778	—	—	—	—	—	—	—	IC9MD	—	—	—	—	—	—	—	OC9MD	0000
PMD6	077A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPI3MD	0000

图注： — = 未实现，读为 0。复位值以十六进制形式显示。

3.2.5 软件堆栈

除了用作工作寄存器外，PIC24F 器件中的 W15 寄存器也可用作软件堆栈指针。堆栈指针总是指向堆栈顶部第一个可供使用的字，从低地址向高地址方向生长。出栈操作指针预减 1，压栈操作指针后加 1，如图 3-4 所示。对于执行任何 CALL 指令时的 PC 压栈，在压栈前，PC 的 MSB 要进行零扩展，从而确保 MSB 始终清零。

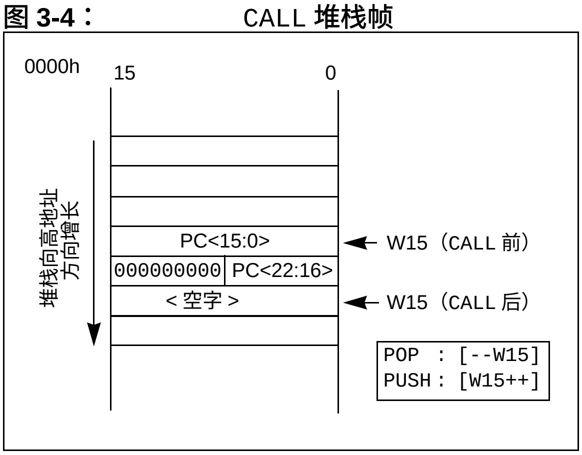
注：	在异常处理期间，在 PC 压入堆栈之前，要先将PC的MSB与SRL寄存器组合在一起。

堆栈指针限制值寄存器（SPLIM）与堆栈指针相关联，它设置堆栈上边界的地址。SPLIM 在复位时不会被初始化。与堆栈指针一样， SPLIM<0> 被强制为 0，因为所有的堆栈操作必须是字对齐的。每当使用 W15 作为源指针或目标指针产生有效地址时，有效地址会与 SPLIM 中的值进行比较。如果堆栈指针（W15）的内容与 SPLIM 寄存器的内容相等，则会执行压栈操作而不产生堆栈错误陷阱，但在随后的压栈操作时将会产生堆栈错误陷阱。因此，例如如果想要在堆栈超过 RAM 中的地址 2000h 时产生堆栈错误陷阱，则需用值 1FFEh 来初始化 SPLIM。

同样，在堆栈指针地址小于 0800h 时产生堆栈指针下溢（堆栈错误）陷阱。这避免了堆栈进入特殊功能寄存器（SFR）空间。

在对 SPLIM 寄存器进行写操作后，不应紧跟使用 W15 进行间接读操作的指令。

图 3-4： CALL 堆栈帧



3.3 程序存储空间与数据存储空间的接口

PIC24F 架构采用 24 位宽的程序空间和 16 位宽的数据空间。该架构也是一种改进的哈佛结构，这意味着数据也能存放在程序空间内。要成功使用程序存储器中的数据，在访问数据时必须确保这两种存储空间中的信息是对齐的。

除了正常执行外， PIC24F 架构还提供了两种可在操作过程中访问程序空间的方法：

- 使用表指令访问程序空间中任意位置的各个字节或字
- 将程序空间的一部分重新映射到数据空间（程序空间可视性）

表指令允许应用程序读写程序存储器中的一小块区域。这一功能对于访问需要随时更新的数据表来说非常理想。也可通过表操作访问一个程序字的所有字节。重映射方法只允许应用程序读数据空间中的大块数据，它非常适合于在一个大的静态数据表中进行查找。这一方式只能访问程序字的低位字。

3.3.1 对程序空间进行寻址

由于数据和程序空间的地址范围分别为 16 位和 24 位，因此需要一个从 16 位数据寄存器创建一个 23 位或 24 位程序地址的方法。方法取决于所采用的接口方式。

对于表操作，使用 8 位表存储器页地址寄存器（TBLPAG）定义程序空间中一个 32K 字的区域。TBLPAG 寄存器的 8 位与 16 位 EA组合形成了一个完整的 24 位程序空间地址。在这种地址形式下，TBLPAG 的最高位用来决定操作是发生在用户存储区（TBLPAG<7> = 0）中还是配置存储区（TBLPAG<7> = 1）中。

对于重映射操作，使用 8 位的程序空间可视性页寄存器（PSVPAG）定义程序空间中的 16K 字页。当 EA 的最高位为 1 时，PSVPAG 的 8 位与 EA 的低 15 位组合形成一个 23 位的程序空间地址。与表操作不同，重映射操作被严格限制在用户存储区中。

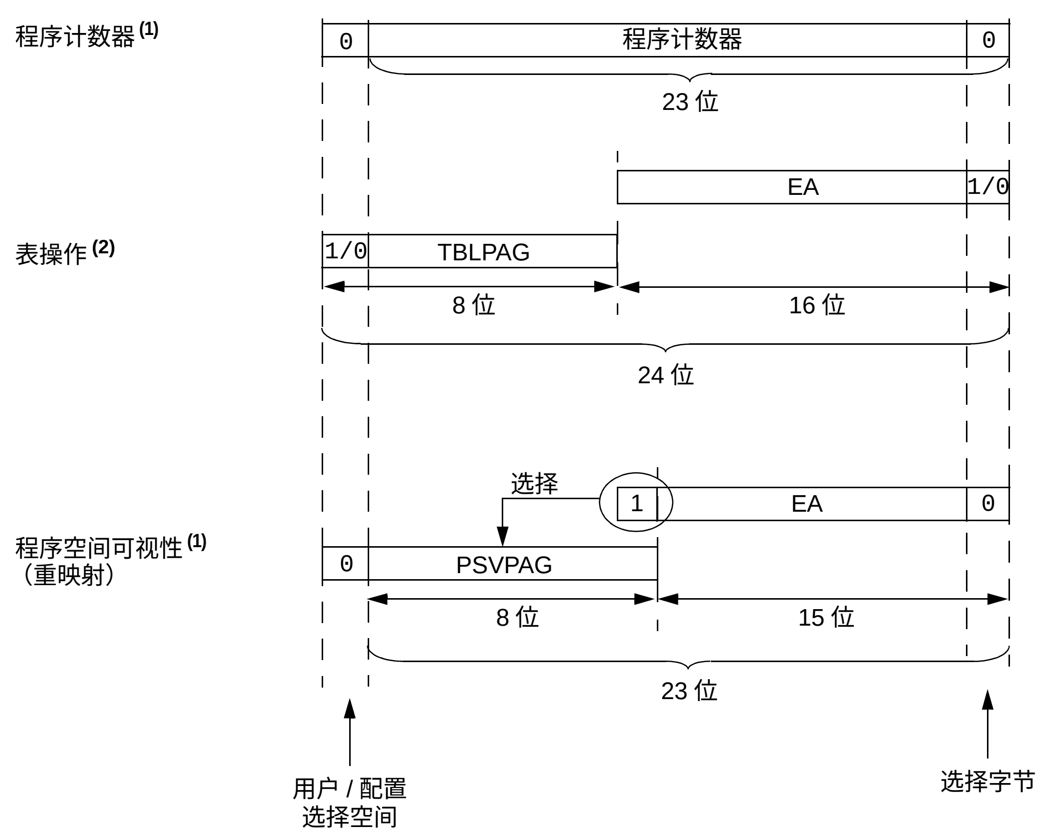
表 3-30 和图 3-5 说明了用数据 EA 创建表操作和重映射访问操作所需 EA 的方法。本文中， P<23:0> 指一个程序空间字，而 D<15:0> 指一个数据空间字。

表 3-30：　程序空间地址构成

访问类型	存取空间	程序空间地址				
		<23>	<22:16>	<15>	<14:1>	<0>
通过指令访问（代码执行）	用户	0	PC<22:1>			0
		0xx xxxx xxxx xxxx xxxx xxx0				
TBLRD/TBLWT（读 / 写字节或字）	用户	TBLPAG<7:0>		数据 EA<15:0>		
		0xxx xxxx		xxxx xxxx xxxx xxxx		
	配置	TBLPAG<7:0>		数据 EA<15:0>		
		1xxx xxxx		xxxx xxxx xxxx xxxx		
程序空间可视性（块重映射 / 读）	用户	0	PSVPAG<7:0>		数据 EA<14:0>[(1)]	
		0	xxxx xxxx		xxx xxxx xxxx xxxx	

注　1：在这种情况下数据 EA<15> 始终为 1，但不用于计算程序空间地址。地址的 bit 15 是 PSVPAG<0>。

图 3-5：　访问程序空间内的数据的地址生成方式



注　1：程序空间地址的最低位始终为 0，其目的是为了使程序和数据空间内的数据保持字对齐。
　　2：表操作不要求字对齐。允许在配置存储空间中执行表读操作。

3.3.2 使用表指令访问程序存储器中的数据

TBLRDL和TBLWTL指令提供了直接读或写程序空间内任何地址的低位字的方法，无需通过数据空间。TBLRDH 和 TBLWTH 指令是可以将一个程序空间字的高 8 位作为数据读写的惟一方法。

对于每个连续的 24 位程序字，PC 的递增量为二。这使得程序存储器地址能够直接映射到数据空间地址。于是，程序存储器可以被看作是两个 16 位字宽的地址空间，它们并排放置，具有相同的地址范围。TBLRDL 和 TBLWTL 访问包含最低有效数据字的空间，TBLRDH 和 TBLWTH 访问包含高数据字节的空间。

提供了两条表指令来对程序空间执行字节或字（16 位）大小的数据读写。读和写都可以采用字节或字操作的形式。

1. TBLRDL（表读低位字）：在字模式下，指令将程序空间位置（P<15:0>）的低字映射到数据地址（D<15:0>）。

   在字节模式中，低位程序字的高字节或低字节被映射到数据地址的低字节中。当字节选择位为 1 时映射高字节；当字节选择位为 0 时映射低字节。

2. TBLRDH（表读高位字）：在字模式下，指令将程序地址（P<23:16>）的整个高字映射到数据地址。注意，D<15:8>为“虚拟”字节，它始终为0。

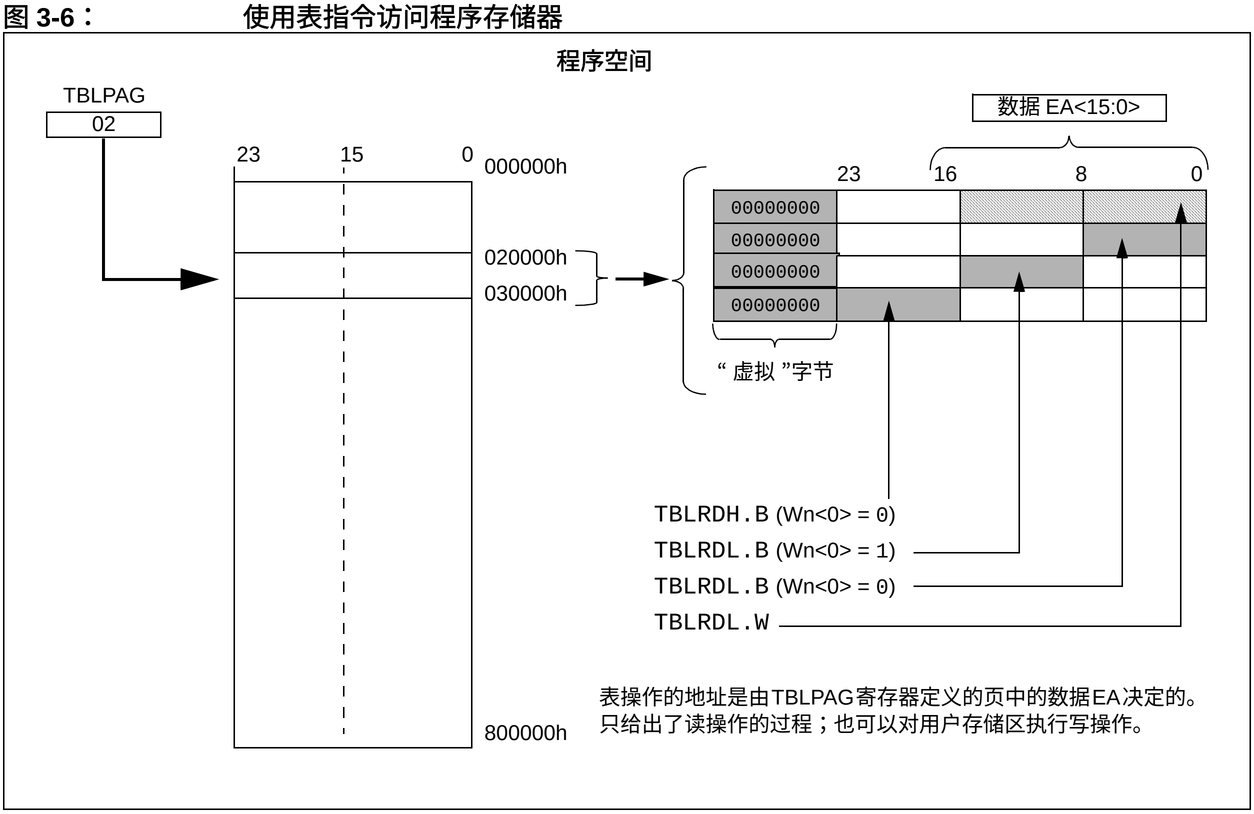
   在字节模式中，程序字的高字节或低字节被映射到数据地址的 D<7:0> 中，这和上一种情况相同。注意当选择高位“虚拟”字节（字节选择位 = 1）时，数据将始终为 0。

TBLWTH 和 TBLWTL 这两条指令采用与读写低位字类似的方式将各个字节或字写入程序空间地址。**第 4.0 节“闪存程序存储器”**对这两条指令的详细操作给出了说明。

对于所有的表操作，要访问程序存储空间的哪个区域是由表存储器页地址寄存器（TBLPAG）决定的。TBLPAG 可以对器件的整个程序存储空间进行寻址，包括用户空间和配置空间。当 TBLPAG<7> = 0 时，表页位于用户存储区中。当 TBLPAG<7> = 1 时，表页位于配置空间中。

> **注：** 仅表读操作可在配置存储空间内执行，且仅在实现的区域，例如器件 ID 中执行。不允许表写操作。

图 3-6： 使用表指令访问程序存储器



3.3.3 使用程序空间可视性来读程序存储器中的数据

可选择将数据空间的高 32 KB 映射到程序空间中的任何 16K 字页中。这样做可以通过数据空间提供对存储的常量数据的透明访问，而无需使用特殊指令（即 TBLRDL/H 指令）。

如果数据空间 EA 的最高位为 1，且通过将 CPU 控制寄存器中的 PSV 位（CORCON<2>）置 1 使能程序空间可视性，就可以通过数据空间访问程序空间。要被映射到数据空间的程序存储空间的位置是由程序空间可视性页地址寄存器（PSVPAG）决定的。这一 8 位寄存器定义了程序空间中 256 个可能的 16K 字页中的任意一个。实际上，PSVPAG 用作程序存储器地址的高 8 位，EA 的 15 位用作低位。注意，对于每个程序存储字 PC 都将递增 2，数据空间地址的低 15 位将直接映射到相应程序空间地址的低 15 位。

将数据读入该区域的指令，需要一个额外的指令周期，因为这类指令需要对程序存储器执行两次数据取操作。

尽管每个数据空间地址，8000h 和更高地址，直接映射到对应的程序存储器地址（见图 3-7），但只使用 24 位程序字的低 16 位来存放数据。任何用作数据存储空间的程序空间地址单元的高 8 位都应编程为 1111 1111 或 0000 0000，以强制为一条 NOP 指令。从而避免了可能出现意外执行这一区域内的代码的情况。

注：	在执行表读写操作时，PSV 访问被暂时禁止。

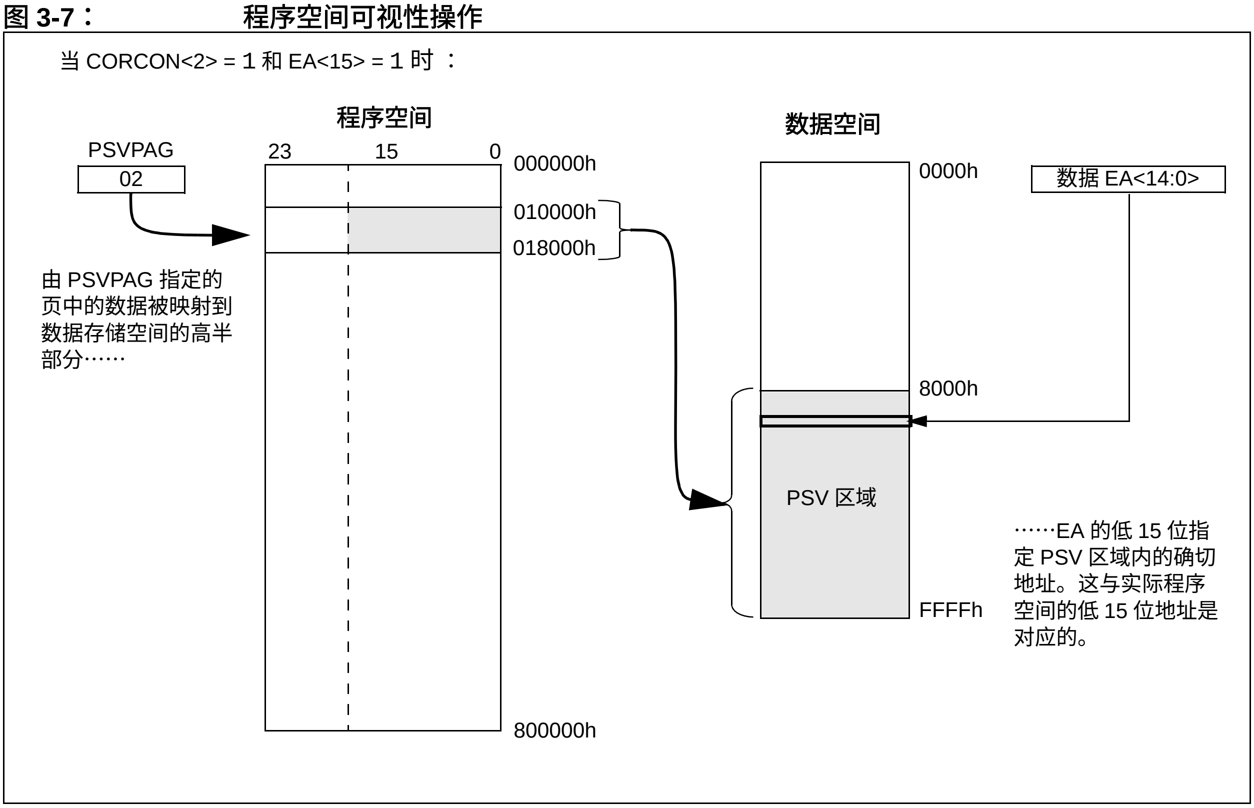
对于使用 PSV 而又在 REPEAT 循环之外执行的指令，MOV 和 MOV.D 指令除了规定的执行时间之外，还需要一个额外的指令周期。其他所有的指令，除了规定的指令执行时间之外，需要两个额外的指令周期。

对于使用 PSV 而又在 REPEAT 循环内执行的指令，下列情况，除了规定的指令执行时间之外，需要两个额外的指令周期：

- 在第一次迭代中执行的指令
- 在最后一次迭代中执行的指令
- 由于中断而退出循环之前执行的指令
- 中断得到处理后而再次进入循环时执行的指令

REPEAT 循环的所有其他迭代，都允许使用 PSV 访问数据的指令在一个周期内执行。

图 3-7： 程序空间可视性操作



4.0 闪存程序存储器

> **注：** 本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的 **“第 4 章 程序存储器”**（DS39715A_CN）。

PIC24FJ256GA110 系列器件包含存储和执行应用程序代码的内部闪存程序存储器。闪存有以下四种编程方式：

- 在线串行编程 （In-Circuit Serial Programming™，ICSP™）
- 运行时自编程 （Run-Time Self-Programming，RTSP）
- JTAG
- 增强型在线串行编程 （增强型 ICSP）

ICSP 允许在最终应用电路中对 PIC24FJ256GA110 系列器件进行串行编程。只需五根线就可以完成编程，它们分别是编程时钟线（PGECx）、编程数据线（PGEDx）、电源线（VDD）、地线（VSS）和主复位线（MCLR）。这允许用户使用未编程器件制造电路板，仅在产品交付前才对单片机进行编程。从而可以将最新版本的固件或定制固件烧写到单片机中。

RTSP 是通过 TBLRD（表读）和 TBLWT（表写）指令完成的。使用 RTSP，用户可以一次将 64 条指令（192 字节）的数据块写入程序存储器，也可以一次擦除 512 条指令（1536 字节）。

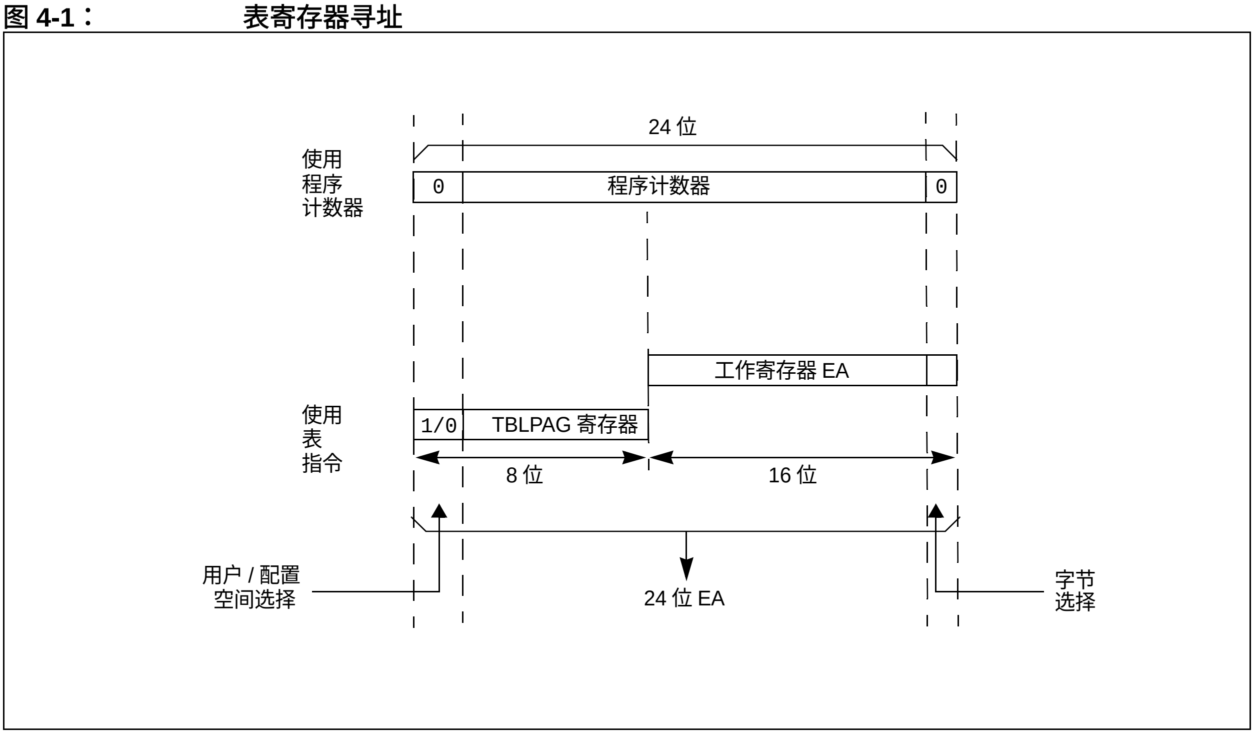
4.1 表指令和闪存编程

闪存的所有编程都是通过表读和表写指令完成的，与使用的方法无关。这些指令允许在器件的正常工作模式下通过数据存储器直接读写程序存储空间。24 位程序存储器目标地址由 TBLPAG<7:0> 位和表指令中指定的 W 寄存器中的有效地址（Effective Address，EA）组成，如图 4-1 所示。

TBLRDL 和 TBLWTL 指令用于读写程序存储空间的 bits<15:0>。TBLRDL 和 TBLWTL 能以字模式或字节模式访问程序存储器。

TBLRDH 和 TBLWTH 指令用于读或写程序存储器的 bits<23:16>。TBLRDH 和 TBLWTH 指令也能以字或字节模式访问程序存储器。

图 4-1： 表寄存器寻址



4.2 RTSP 工作原理

PIC24F 闪存程序存储器阵列以 64 条指令或 192 字节的行为单位构成。RTSP 允许用户一次擦除八行（512 条指令）的块并一次编程一行。它还可以编程单字。

8行擦除块和单行写入块都是边界对齐的，从程序存储器起始地址开始，分别到 1536 字节边界和 192 字节边界。

用 TBLWT 指令将数据写入程序存储器时，数据并未直接写入存储器。而是存储在保持锁存器中，直到执行编程序列。

可以执行任何数量的 TBLWT 指令，写操作都将成功执行。但是，需要 64 条 TBLWT 指令来写整行存储器。

要确保在写操作期间没有数据被改动，应将所有未使用的地址编程为FFFFFFh。这是因为保持锁存器复位为未知状态，因此当地址处于复位状态时，就可能改写未被重写的行的单元。

RTSP 编程的基本步骤是先建立一个表指针，然后执行一系列 TBLWT 指令将数据装入缓冲器。通过设置 NVMCON 寄存器的控制位来执行编程。

可按任何顺序装入数据，且在执行写操作之前可以对保持寄存器进行多次写操作。但后续写操作将覆盖先前的所有写操作。

注：	不推荐多次写入一页而不擦除其内容。

因为只写缓冲器，因此所有表写操作都是单字写操作（2 个指令周期）。编程每行都需要一个编程周期。

4.3 JTAG 操作

PIC24F 系列支持 JTAG 编程和边界扫描。边界扫描可以通过验证引脚到 PCB 的连通性改进制造工艺。编程是通过支持串行向量格式（Serial Vector Format，SVF）的工业标准 JTAG 编程器来执行的。

4.4 增强型在线串行编程

增强型在线串行编程使用称为编程执行程序的片上自举程序管理编程过程。通过使用 SPI 数据帧格式，编程执行程序可以擦除、编程和验证程序存储器。如需了解更多有关增强型 ICSP 的信息，请参见编程规范。

4.5 控制寄存器

有两个用于读写闪存程序存储器的 SFR：NVMCON 和 NVMKEY。

NVMCON 寄存器（寄存器 4-1）控制要擦除的块、要编程的存储器类型以及编程周期的开始时间。

NVMKEY 是只写寄存器，用于写保护。要启动编程或擦除过程，用户必须把 55h 和 AAh 连续写入 NVMKEY 寄存器。更多详细信息，请参见**第 4.6 节 “ 编程操作 ”**。

4.6 编程操作

在 RTSP 模式下，对内部闪存进行编程或擦除需要完整的编程过程。在编程或擦除操作期间，处理器将停止（等待），直到操作完成。将 WR 位（NVMCON<15>）置 1 启动操作，当操作完成时 WR 位会自动清零。

寄存器 4-1： NVMCON：闪存控制寄存器

R/SO-0(1)	R/W-0(1)	R/W-0(1)	U-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0(1)	U-0	U-0	R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0(1)
—	ERASE	—	—	NVMOP3(2)	NVMOP2(2)	NVMOP1(2)	NVMOP0(2)
bit 7							bit 0

图注：	SO = 只置 1 位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **WR：**写控制位 (1)
1 = 启动闪存编程或擦除操作。操作是自定时的，一旦操作完成，此位即由硬件清零。
0 = 编程或擦除操作完成，并处于停止状态

bit 14 **WREN：**写使能位 (1)
1 = 使能闪存编程 / 擦除操作
0 = 禁止闪存编程 / 擦除操作

bit 13 **WRERR：**写序列错误标志位 (1)
1 = 尝试执行错误的编程或擦除序列或发生终止 （任何试图置 1 WR 位的操作都会自动置 1 此位）
0 = 编程或擦除操作正常完成

bit 12-7 **未实现：**读为 0

bit 6 **ERASE：**擦除 / 编程使能位 (1)
1 = 在下一条写命令执行由 NVMOP3:NVMOP0 指定的擦除操作
0 = 在下一条写命令执行由 NVMOP3:NVMOP0 指定的编程操作

bit 5-4 **未实现：**读为 0

bit 3-0 **NVMOP3:NVMOP0：**NVM 操作选择位 (1,2)
1111 = 存储器批量擦除操作 （ERASE = 1）或无操作 （ERASE = 0） (3)
0011 = 存储器字编程操作 （ERASE = 0）或无操作 （ERASE = 1）
0010 = 存储器页擦除操作 （ERASE = 1）或无操作 （ERASE = 0）
0001 = 存储器行编程操作 （ERASE = 0）或无操作 （ERASE = 1）

注 1： 只能在上电复位时复位这些位。
2： NVMOP3:NVMOP0 的所有其他组合均未实现。
3： 仅在 ICSP™ 模式下可用。请参见器件编程规范。

4.6.1 闪存程序存储器的编程算法

用户一次可编程闪存程序存储器的一行。要实现此操作，有必要擦除包含该行在内的一个 8 行大小的块。一般步骤如下：

1. 读程序存储器中的八行（512 条指令），并将其存储在数据 RAM 中。
2. 用需要的新数据更新 RAM 中对应的程序数据。
3. 擦除块（见例 4-1）：
   a) 将 NVMOP 位（NVMCON<3:0>）设置为 `0010`，以配置为块擦除操作。将 ERASE（NVMCON<6>）和 WREN（NVMCON<14>）位置 1。
   b) 将要擦除块的起始地址写入 TBLPAG 和 W 寄存器。
   c) 将 55h 写入 NVMKEY。
   d) 将 AAh 写入 NVMKEY。
   e) 将 WR 位（NVMCON<15>）置 1。启动擦除周期，CPU 停止，等待擦除周期完成。擦除操作完成时，WR 位自动清零。
4. 将数据 RAM 的前 64 条指令写入程序存储器缓冲器（见例 4-1）。
5. 将程序块写入闪存：
   a) 将 NVMOP 位设置为 `0001`，以配置为行编程操作。将 ERASE 位清零，将 WREN 位置 1。
   b) 将 55h 写入 NVMKEY。
   c) 将 AAh 写入 NVMKEY。
   d) 将 WR 位置 1。启动编程周期，CPU 停止工作等待写周期完成。当写闪存的操作完成时，WR 位自动清零。
6. 将 TBLPAG 的值递增 `1`，使用数据 RAM 中下一个 64 条指令的块重复步骤 4 和 5，直到将所有 512 条指令写回闪存。

为了防止意外操作，必须向 NVMKEY 写入写启动序列从而允许执行任何擦除或编程操作。执行了编程命令以后，用户必须等待一定的编程时间，直到编程操作完成。编程序列开始后紧跟的两条指令必须为 `NOP`，如例 4-3 所示。

例 4-1： 擦除程序存储器块

```
; Set up NVMCON for block erase operation
       MOV     #0x4042, W0                  ;
       MOV     W0, NVMCON                   ; Initialize NVMCON
; Init pointer to row to be ERASED
       MOV     #tblpage(PROG_ADDR), W0      ;
       MOV     W0, TBLPAG                   ; Initialize PM Page Boundary SFR
       MOV     #tbloffset(PROG_ADDR), W0    ; Initialize in-page EA[15:0] pointer
       TBLWTL  W0, [W0]                     ; Set base address of erase block
       DISI    #5                           ; Block all interrupts with priority <7
                                            ; for next 5 instructions
       MOV     #0x55, W0
       MOV     W0, NVMKEY                   ; Write the 55 key
       MOV     #0xAA, W1                    ;
       MOV     W1, NVMKEY                   ; Write the AA key
       BSET    NVMCON, #WR                  ; Start the erase sequence
       NOP                                  ; Insert two NOPs after the erase
       NOP                                  ; command is asserted
```


例 4-2： 装载写缓冲器

```
; Set up NVMCON for row programming operations
       MOV      #0x4001, W0                     ;
       MOV      W0, NVMCON                      ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV      #0x0000, W0                     ;
       MOV      W0, TBLPAG                      ; Initialize PM Page Boundary SFR
       MOV      #0x6000, W0                     ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       MOV      #LOW_WORD_0, W2                 ;
       MOV      #HIGH_BYTE_0, W3                ;
       TBLWTL   W2, [W0]                        ; Write PM low word into program latch
       TBLWTH   W3, [W0++]                      ; Write PM high byte into program latch
; 1st_program_word
       MOV      #LOW_WORD_1, W2                 ;
       MOV      #HIGH_BYTE_1, W3                ;
       TBLWTL   W2, [W0]                        ; Write PM low word into program latch
       TBLWTH   W3, [W0++]                      ; Write PM high byte into program latch
;  2nd_program_word
       MOV      #LOW_WORD_2, W2                 ;
       MOV      #HIGH_BYTE_2, W3                ;
       TBLWTL   W2, [W0]                        ; Write PM low word into program latch
       TBLWTH   W3, [W0++]                      ; Write PM high byte into program latch
       •
       •
       •
; 63rd_program_word
       MOV      #LOW_WORD_31, W2                ;
       MOV      #HIGH_BYTE_31, W3               ;
       TBLWTL   W2, [W0]                        ; Write PM low word into program latch
       TBLWTH   W3, [W0]                        ; Write PM high byte into program latch
```

例 4-3： 启动编程序列

```
       DISI    #5                         ; Block all interrupts with priority <7
                                          ; for next 5 instructions
       MOV     #0x55, W0
       MOV     W0, NVMKEY                 ; Write the 55 key
       MOV     #0xAA, W1                  ;
       MOV     W1, NVMKEY                 ; Write the AA key
       BSET    NVMCON, #WR                ; Start the erase sequence
       BTSC    NVMCON, #15                ; and wait for it to be
       BRA     $-2                        ; completed
```

4.6.2 编程闪存程序存储器的一个单字

若已擦除了一个闪存单元，则可用表写指令对此单元进行编程以将一个指令字 （24 位）写入写锁存器。将闪存地址的 8 个最高字节装入 TBLPAG 寄存器。TBLWTL 和 TBLWTH 指令将所需数据写入写锁存器，并指定要写入的程序存储器地址的低 16 位。要将 NVMCON 寄存器配置为字写操作，应将NVMOP位（NVMCON<3:0>）设置为 0011。通过执行解锁序列并置 1 WR 位来执行此写操作 （见例 4-4）。

例 4-4： 编程闪存程序存储器的一个单字

```
; Setup a pointer to data Program Memory
    MOV     #tblpage(PROG_ADDR), W0       ;
    MOV     W0, TBLPAG                    ;Initialize PM Page Boundary SFR
    MOV     #tbloffset(PROG_ADDR), W0     ;Initialize a register with program memory address

    MOV     #LOW_WORD_N, W2               ;
    MOV     #HIGH_BYTE_N, W3              ;
    TBLWTL  W2, [W0]                      ; Write PM low word into program latch
    TBLWTH  W3, [W0++]                    ; Write PM high byte into program latch

; Setup NVMCON for programming one word to data Program Memory
    MOV     #0x4003, W0                   ;
    MOV     W0, NVMCON                    ; Set NVMOP bits to 0011

    DISI    #5                            ; Disable interrupts while the KEY sequence is written
    MOV     #0x55, W0                     ; Write the key sequence
    MOV     W0, NVMKEY
    MOV     #0xAA, W0
    MOV     W0, NVMKEY
    BSET    NVMCON, #WR                   ; Start the write cycle
```


5.0 复位

注：	本数据手册总结了 PIC24F 系列器件的功能。但不应当作全面的参考资料。更多信息，请参见《PIC24F 系列参考手册》中的 **“第 7 章 复位”**（DS39712A_CN）。

复位模块结合了所有复位资源并控制器件的主复位信号 SYSRST。以下所列是器件复位源：

- POR：上电复位
- MCLR：引脚复位
- SWR：RESET 指令
- WDT：看门狗定时器复位
- BOR：欠压复位
- CM：配置失配复位
- TRAPR：陷阱冲突复位
- IOPUWR：非法操作码复位
- UWR：未初始化的 W 寄存器复位

图 5-1 所示为复位模块的简化框图。

任何激活的复位源都会激活SYSRST信号。许多与CPU 和外设有关的寄存器被强制为一个已知的复位状态。多数寄存器都不受复位影响；在上电复位时寄存器状态未知，而在其他复位时寄存器状态不变。

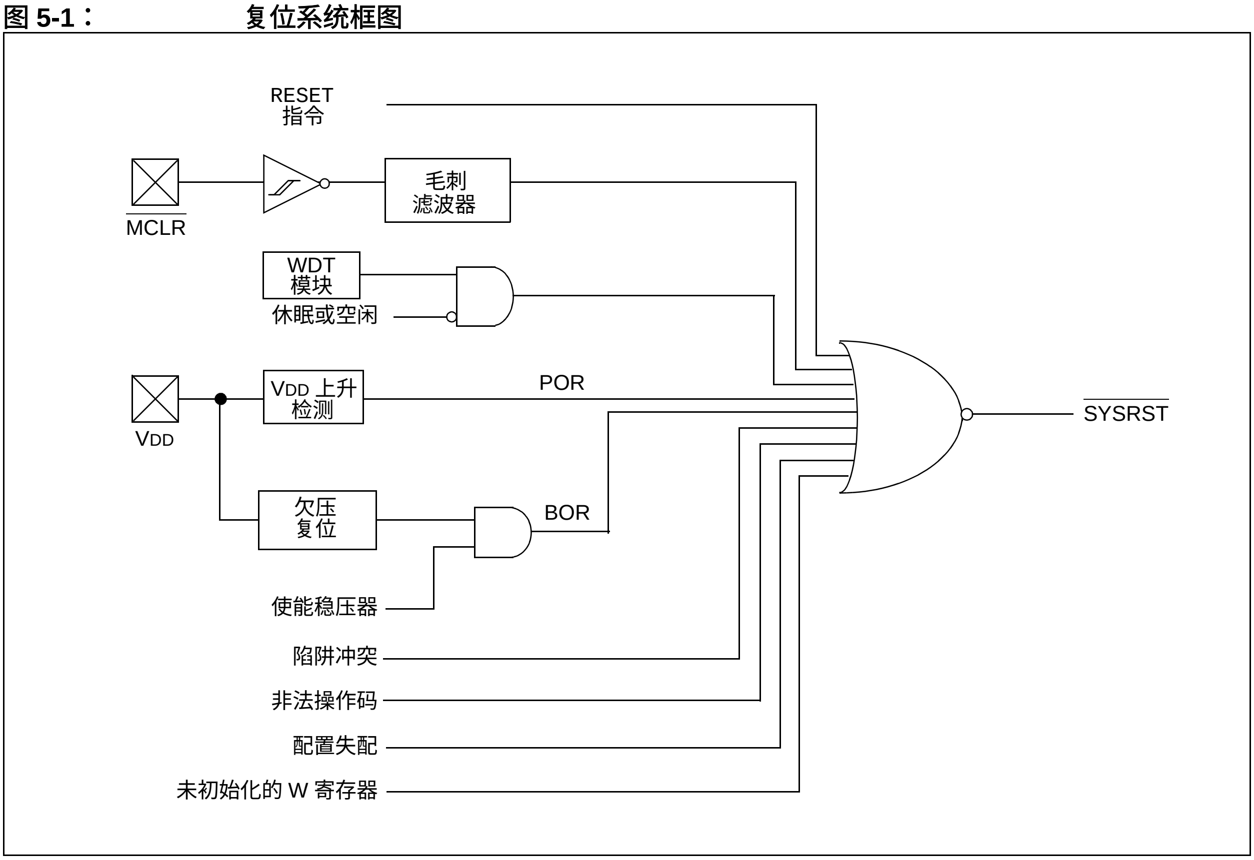
注：	有关寄存器复位状态的信息，请参见本手册中的特定外设或 CPU 章节。

任何类型的器件复位都将RCON寄存器中相应的状态位置 1 以显示复位类型 （见寄存器 5-1）。上电复位会清零除 BOR 和 POR 位 （RCON<1:0>）外的所有位，而 BOR 和 POR 位会被置 1。用户能在代码执行过程中的任何时间将任何位置 1 或清零。 RCON 位仅用作状态位。用软件将特定的状态位置 1 不会导致器件发生复位。

RCON寄存器也有一些与看门狗定时器和器件节能状态相关的位。本手册的其他章节中将讨论这些位的功能。

注：	RCON 寄存器中的状态位应该在被读取后清零，这样下一个器件复位后 RCON 寄存器的值才有意义。

图 5-1： 复位系统框图



寄存器 5-1：　RCON：复位控制寄存器 (1)

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
TRAPR	IOPUWR	—	—	—	—	CM	VREGS
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
EXTR	SWR	SWDTEN(2)	WDTO	SLEEP	IDLE	BOR	POR
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知位

bit 15　**TRAPR：**陷阱复位标志位
1 = 发生了陷阱冲突复位
0 = 未发生陷阱冲突复位

bit 14　**IOPUWR：**非法操作码或访问未初始化的 W 寄存器复位标志位
1 = 非法操作码检测、非法地址模式或未初始化的 W 寄存器用作地址指针而导致的复位
0 = 没有发生非法操作码或未初始化的 W 寄存器复位

bit 13-10　**未实现：**读为 0

bit 9　**CM ：**配置字不匹配复位标志位
1 = 发生了配置字不匹配复位
0 = 未发生配置字不匹配复位

bit 8　**VREGS：**稳压器待机使能位
1 = 稳压器在休眠期间保持工作状态
0 = 稳压器在休眠期间进入待机状态

bit 7　**EXTR：**外部复位 （$\overline{MCLR}$）引脚位
1 = 发生了主复位 （引脚） 复位
0 = 未发生主复位 （引脚） 复位

bit 6　**SWR：**软件复位 （指令）标志位
1 = 执行了 RESET 指令
0 = 未执行 RESET 指令

bit 5　**SWDTEN：**软件使能 / 禁止 WDT 位 (2)
1 = 使能 WDT
0 = 禁止 WDT

bit 4　**WDTO：**看门狗定时器超时标志位
1 = WDT 发生了超时
0 = WDT 未发生超时

bit 3　**SLEEP：**从休眠模式唤醒的标志位
1 = 器件处于休眠模式
0 = 器件未处于休眠模式

bit 2　**IDLE：**从空闲模式唤醒的标志位
1 = 器件处于空闲模式
0 = 器件未处于空闲模式

bit 1　**BOR：**欠压复位标志位
1 = 发生了欠压复位。注意 BOR 在上电复位后也被置 1。
0 = 未发生欠压复位

bit 0　**POR：**上电复位标志位
1 = 发生了上电复位
0 = 未发生上电复位

注　1： 所有复位状态位都可以用软件置 1 或清零。用软件将这些位中的一位置 1 不会导致器件复位。
　　2： 如果 FWDTEN 配置位为 1 （未编程），不管 SWSTEN 位是否置 1，总是使能 WDT。

表 5-1：　　复位标志位操作

标志位	置 1 事件	清零事件
TRAPR （RCON<15>）	陷阱冲突事件	POR
IOPUWR （RCON<14>）	非法操作码或访问未初始化的 W 寄存器	POR
CM （RCON<9>）	配置失配复位	POR
EXTR （RCON<7>）	$\overline{\text{MCLR}}$ 复位	POR
SWR （RCON<6>）	`RESET` 指令	POR
WDTO （RCON<4>）	看门狗定时器超时	`PWRSAV` 指令和 POR
SLEEP （RCON<3>）	`PWRSAV #SLEEP` 指令	POR
IDLE （RCON<2>）	`PWRSAV #IDLE` 指令	POR
BOR （RCON<1>）	POR 和 BOR	—
POR （RCON<0>）	POR	—

注：　所有复位标志位均可由用户软件置 1 或清零。

5.1　复位时的时钟源选择

表 5-2 所示为允许时钟切换时，器件复位时的系统时钟源的选择。如果禁止了时钟切换功能，则总是根据振荡器配置位选择系统时钟源。更多详细信息请参见**第 7.0 节 " 振荡器配置 "**。

表 5-2：　　振荡器选择－复位类型（使能时钟切换）

复位类型	时钟源的确定
POR	FNOSC 配置位（CW2<10:8>）
BOR	
$\overline{\text{MCLR}}$	COSC 控制位（OSCCON<14:12>）
WDTO	
SWR	

5.2　器件复位时间

表 5-3 总结了各种器件复位的复位时间。注意在 POR 延时和 PWRT 延时结束后会发出系统复位信号 SYSRST。

器件实际开始执行代码的时间还取决于系统的振荡器延时，它包括振荡器起振定时器（Oscillator Start-up Timer，OST）和 PLL 锁定时间。OST 和 PLL 锁定时间与相应的 SYSRST 延时同时发生。

FSCM 延时决定在 $\overline{\text{SYSRST}}$ 信号发出后 FSCM 开始监视系统时钟源的时间。

表 5-3： 各种器件复位的复位延迟时间

复位类型	时钟源	$\overline{\text{SYSRST}}$ 延时	系统时钟延时	FSCM 延时	注
POR	EC、FRC、FRCDIV 和 LPRC	TPOR + TSTARTUP + TRST	—	—	**1, 2, 3**
	ECPLL 和 FRCPLL	TPOR + TSTARTUP + TRST	TLOCK	TFSCM	**1, 2, 3, 5, 6**
	XT、 HS 和 SOSC	TPOR + TSTARTUP + TRST	TOST	TFSCM	**1, 2, 3, 4, 6**
	XTPLL 和 HSPLL	TPOR + TSTARTUP + TRST	TOST + TLOCK	TFSCM	**1, 2, 3, 4, 5, 6**
BOR	EC、FRC、FRCDIV 和 LPRC	TSTARTUP + TRST	—	—	**2, 3**
	ECPLL 和 FRCPLL	TSTARTUP + TRST	TLOCK	TFSCM	**2, 3, 5, 6**
	XT、 HS 和 SOSC	TSTARTUP + TRST	TOST	TFSCM	**2, 3, 4, 6**
	XTPLL 和 HSPLL	TSTARTUP + TRST	TOST + TLOCK	TFSCM	**2, 3, 4, 5, 6**
$\overline{\text{MCLR}}$	任何时钟	TRST	—	—	**3**
WDT	任何时钟	TRST	—	—	**3**
软件	任何时钟	TRST	—	—	**3**
非法操作码	任何时钟	TRST	—	—	**3**
未初始化的W寄存器	任何时钟	TRST	—	—	**3**
陷阱冲突	任何时钟	TRST	—	—	**3**

注 1： TPOR = 上电复位延时 （标称值为 10 µs）。
2： 如果使能片上稳压器，则 TSTARTUP = TVREG （标称值为 10 µs）；如果禁止片上稳压器，则 TSTARTUP = TPWRT （标称值为 64 ms）。
3： TRST = 内部状态复位时间 （标称值为 32 µs）。
4： TOST = 振荡器起振定时器延时。 10 位计数器计满 1024 个振荡器周期后，才将振荡器时钟释放给系统使用。
5： TLOCK = PLL 锁定时间。
6： TFSCM = 故障保护时钟监视器延时 （标称值为 100 µs）。

5.2.1 POR 和长振荡器起振时间

振荡器起振电路及其相关的延时定时器与上电时发生的器件复位延时没有关系。某些晶振电路（尤其是低频晶振）的起振时间会相对较长。因此，在 SYSRST 发出后，可能会发生以下一个或多个情况：

- 振荡电路未开始振荡。
- 振荡器起振定时器尚未超时 （如果使用了晶振）。
- PLL 还未锁定 （如果使用了 PLL）。

在有效时钟源供系统使用前，器件不会开始执行代码。因此，在确定复位延迟时间时，还须考虑振荡器和 PLL 启动延时。

5.2.2 故障保护时钟监视器 （FSCM）和器件复位

如果使能了 FSCM，它将在 $\overline{\text{SYSRST}}$ 发出时开始监视系统时钟源。如果此时没有可用的有效时钟源，器件将自动切换到 FRC 振荡器，并且用户可以在陷阱服务程序中切换到所需的晶体振荡器。

5.2.2.1 晶振和 PLL 时钟源的 FSCM 延时

当系统时钟源由晶体振荡器和 / 或 PLL 提供时，POR 和 PWRT 延时后将自动插入一小段延时 T_{FSCM}。在此延时结束后，FSCM 才开始监视系统时钟源。FSCM 延迟时间标称值为 100 μs，为振荡器和 / 或 PLL 达到稳定提供更多的时间。在多数情况下，当禁止了 PWRT 时，FSCM 延时会防止在器件复位时发生振荡器故障陷阱。

5.3 特殊功能寄存器的复位状态

大部分与 PIC24F CPU 和外设有关的特殊功能寄存器会在器件复位时复位为某个特定值。 SFR 是按其外设或 CPU 功能分组的，其复位值在本手册的各部分中有说明。

除了四个寄存器之外，其他 SFR 的复位值都不受复位类型的影响。复位控制寄存器 RCON 的复位值取决于器件复位的类型。振荡器控制寄存器OSCCON的复位值取决于复位的类型和闪存配置字 2 （CW2）中 FNOSC 位的编程值（见表 5-2）。RCFGCAL 和 NVMCON 寄存器只受 POR 影响。

注 ：

6.0 中断控制器

> **注：** 本数据手册总结了 PIC24F 系列器件的功能。但是不包括全部参考资料。更多信息，请参见《PIC24F 系列参考手册》中的 " **第 8 章 中断** "（DS39707A_CN）。

PIC24F 中断控制器将诸多外设中断请求信号缩减到一个到 PIC24F CPU 的中断请求信号。该控制器具有以下特性：

- 最多 8 个处理器异常和软件陷阱
- 7 个用户可选择的优先级
- 具有最多 118 个向量的中断向量表 （Interrupt Vector Table， IVT）
- 每个中断或异常源都有唯一的向量
- 指定的用户优先级中的固定优先级
- 用于支持调试的备用中断向量表 （Alternate Interrupt Vector Table， AIVT）
- 固定的中断进入和返回延时

6.1 中断向量表

中断向量表 （IVT）如图 6-1 所示。 IVT 位于程序存储器中，起始单元地址是 000004h。IVT 包含 126 个向量，由 8 个不可屏蔽的陷阱向量和最多 118 个中断源组成。一般来说，每个中断源都有自己的中断向量。每个中断向量都包含 24 位宽的地址。每个中断向量存储单元中设置的值是其对应的中断服务程序 （Interrupt Service Routine， ISR）的起始地址。

中断向量有一个自然优先级；也就是说每个中断向量的优先级与其在向量表中的位置有关。如果其他方面都相同，较低地址的中断向量具有较高的自然优先级。例如，与向量 0 相关的中断比任何其他向量地址的中断具有更高的自然优先级。

PIC24FJ256GA110 系列器件实现有不可屏蔽陷阱和唯一的中断。表 6-1 和表 6-2 对此做了总结。

6.1.1 备用中断向量表

如图 6-1 所示，备用中断向量表 （AIVT）位于 IVT 之后。ALTIVT 控制位（INTCON2<15>）控制对 AIVT 的访问。如果 ALTIVT 位置 1，所有中断和异常处理将使用备用向量而不是默认的向量。备用向量与默认向量的构成方式相同。

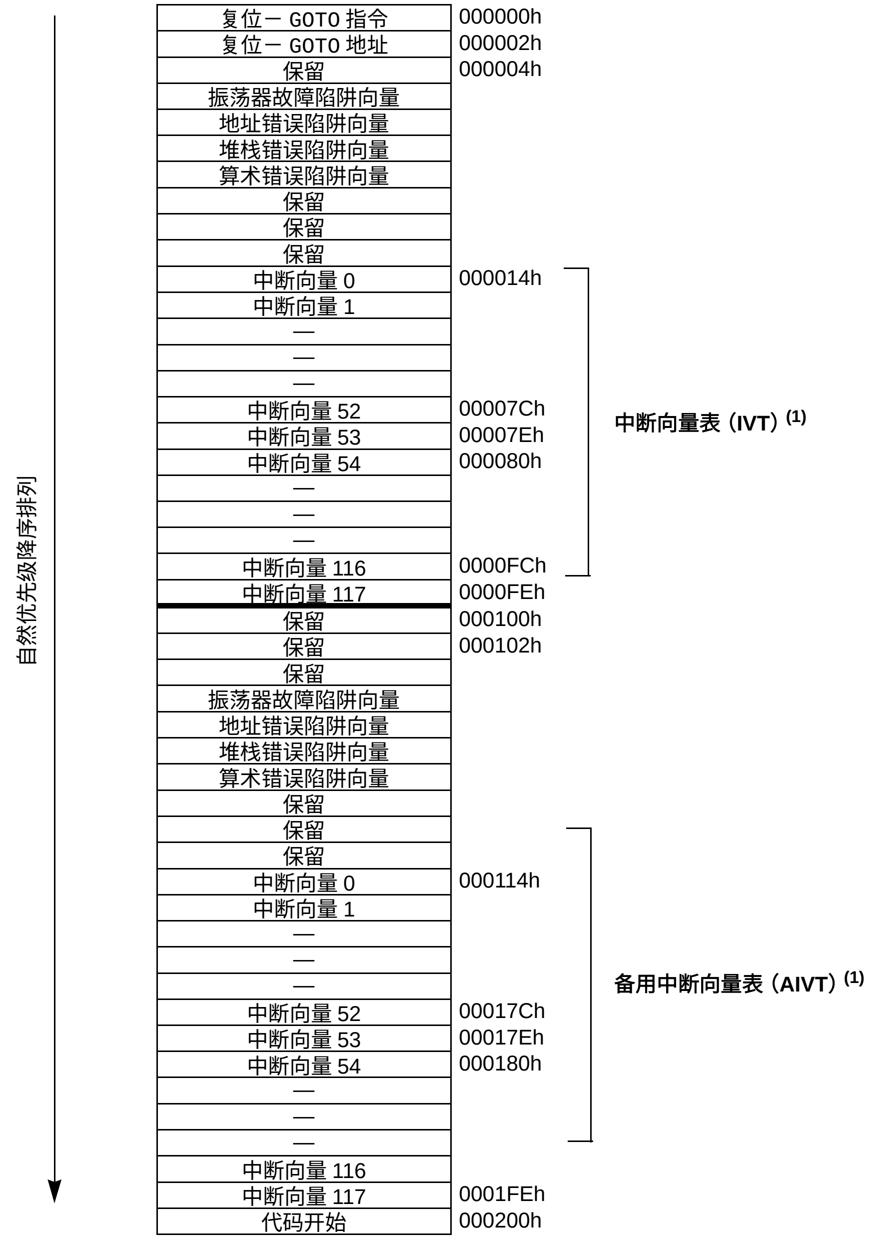
AIVT支持仿真和调试功能，它提供了一种不需要将中断向量再编程就可以在应用程序和支持环境之间切换的方法。此特性也支持运行时在不同应用程序之间切换以便评估各种软件算法。如果不需要 AIVT，应该用 IVT 中使用的地址设置 AIVT。

6.2 复位顺序

由于复位过程中不涉及到中断控制器，所以器件复位并不是真的异常。作为对复位的响应， PIC24F 器件清零其寄存器，同时强制 PC 为零。然后，单片机从单元 000000h 开始执行程序。用户在复位地址上编写 `GOTO` 指令会使程序执行再次定位到相应的起始程序。

> **注：** 应该使用包含 RESET 指令的默认中断处理程序的地址编程 IVT 和 AIVT 中所有未实现或未使用的向量存储单元。

图 6-1：　PIC24F 中断向量表



注　1：有关中断向量列表的信息，请参见表 6-2。

表 6-1：　陷阱向量细节

向量编号	IVT 地址	AIVT 地址	陷阱源
0	000004h	000104h	保留
1	000006h	000106h	振荡器故障
2	000008h	000108h	地址错误
3	00000Ah	00010Ah	堆栈错误
4	00000Ch	00010Ch	算术错误
5	00000Eh	00010Eh	保留
6	000010h	000110h	保留
7	000012h	0001172h	保留

表 6-2： 实现的中断向量

中断源	向量编号	IVT 地址	AIVT 地址	中断位位置		
				标志	允许	优先级
ADC1 转换完成	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
比较器事件	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC 发生器	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
CTMU 事件	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
外部中断 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
外部中断 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
外部中断 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
外部中断 3	53	00007Eh	00017Eh	IFS3<5>	IEC3<5>	IPC13<6:4>
外部中断 4	54	000080h	000180h	IFS3<6>	IEC3<6>	IPC13<10:8>
I2C1 主事件	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 从事件	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 主事件	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 从事件	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
I2C3 主事件	85	0000BEh	0001BEh	IFS5<5>	IEC5<5>	IPC21<6:4>
I2C3 从事件	84	0000BCh	0001BCh	IFS5<4>	IEC5<4>	IPC21<2:0>
输入捕捉 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
输入捕捉 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
输入捕捉 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
输入捕捉 4	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
输入捕捉 5	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
输入捕捉 6	40	000064h	000164h	IFS2<8>	IEC2<8>	IPC10<2:0>
输入捕捉 7	22	000040h	000140h	IFS1<6>	IEC1<6>	IPC5<10:8>
输入捕捉 8	23	000042h	000142h	IFS1<7>	IEC1<7>	IPC5<14:12>
输入捕捉 9	93	0000CEh	0001CEh	IFS5<13>	IEC5<13>	IPC23<6:4>
输入状态变化通知	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
LVD 低电压检测	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
输出比较 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
输出比较 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
输出比较 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
输出比较 4	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
输出比较 5	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
输出比较 6	42	000068h	000168h	IFS2<10>	IEC2<10>	IPC10<10:8>
输出比较 7	43	00006Ah	00016Ah	IFS2<11>	IEC2<11>	IPC10<14:12>
输出比较 8	44	00006Ch	00016Ch	IFS2<12>	IEC2<12>	IPC11<2:0>
输出比较 9	92	0000CCh	0001CCh	IFS5<12>	IEC5<12>	IPC23<2:0>
并行主端口	45	00006Eh	00016Eh	IFS2<13>	IEC2<13>	IPC11<6:4>
实时时钟 / 日历	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>
SPI1 错误	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 事件	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI2 错误	32	000054h	000154h	IFS2<0>	IEC2<0>	IPC8<2:0>
SPI2 事件	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>
SPI3 错误	90	0000C8h	0001C8h	IFS5<10>	IEC5<10>	IPC22<10:8>
SPI3 事件	91	0000CAh	0001CAh	IFS5<11>	IEC5<11>	IPC22<14:12>

表 6-2： 实现的中断向量 （续）

中断源	向量编号	IVT 地址	AIVT 地址	中断位位置		
				标志	允许	优先级
Timer1	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>
Timer2	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>
Timer3	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>
Timer4	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>
Timer5	28	00004Ch	00014Ch	IFS1<12>	IEC1<12>	IPC7<2:0>
UART1 错误	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>
UART1 接收器	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>
UART1 发送器	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>
UART2 错误	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>
UART2 接收器	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>
UART2 发送器	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>
UART3 错误	81	0000B6h	0001B6h	IFS5<1>	IEC5<1>	IPC20<6:4>
UART3 接收器	82	0000B8h	0001B8h	IFS5<2>	IEC5<2>	IPC20<10:8>
UART3 发送器	83	0000BAh	0001BAh	IFS5<3>	IEC5<3>	IPC20<14:12>
UART4 错误	87	0000C2h	0001C2h	IFS5<7>	IEC5<7>	IPC21<14:12>
UART4 接收器	88	0000C4h	0001C4h	IFS5<8>	IEC5<8>	IPC22<2:0>
UART4 发送器	89	0000C6h	0001C6h	IFS5<9>	IEC5<9>	IPC22<6:4>

6.3 中断控制和状态寄存器

PIC24FJ256GA110系列器件一共有36个用于中断控制器的寄存器：

- INTCON1
- INTCON2
- IFS0 至 IFS5
- IEC0 至 IEC5
- IPC0 至 IPC23 （除 IPC14 和 IPC17 之外）

INTCON1 和 INTCON2 控制全局中断。INTCON1 包含中断嵌套禁止 （NSTDIS）位以及处理器陷阱源的控制和状态标志。 INTCON2 寄存器控制外部中断请求信号行为和备用中断向量表的使用。

IFSx 寄存器包含所有中断请求标志。每个中断源都有一个状态位，由各自的外设或外部信号置 1，而由软件清零。

IECx寄存器包含所有的中断允许位。这些控制位用于单独允许外设或外部信号中断。

IPCx寄存器用于设置每个中断源的中断优先级。可以给用户中断源分配八个优先级之一。

中断源按表6-2中的向量编号分配给IFSx、IECx和IPCx寄存器。例如， INT0 （外部中断 0）具有一个向量编号，自然优先级为 0。所以 INT0IF 状态位在 IFS0<0> 中， INT0IE 允许位在 IEC0<0> 中， INT0IP<2:0> 优先级位在 IPC0 最初的位置 （IPC0<2:0>）中。

尽管两个 CPU 控制寄存器都不是中断控制硬件的特定组成部分，但它们仍包含控制中断功能的位。ALU 状态寄存器 （SR）包含 IPL2:IPL0 位 （SR<7:5>）。这些位用于指示当前 CPU 的中断优先级。用户可以通过写入 IPL 位更改 CPU 的当前优先级。

CORCON 寄存器包含 IPL3 位，这个位与 IPL2:IPL0 位一起表示 CPU 的当前优先级。 IPL3 是一个只读位，所以用户软件不能屏蔽陷阱事件。

在下面各页中的寄存器 6-1 到寄存器 6-38 说明了所有的中断寄存器。

寄存器 6-1： SR：ALU 状态寄存器 （CPU 中）

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0
—	—	—	—	—	—	—	DC(1)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2(2,3)	IPL1(2,3)	IPL0(2,3)	RA(1)	N(1)	OV(1)	Z(1)	C(1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 **IPL2:IPL0：**CPU 中断优先级状态位 (2,3)
111 = CPU 中断优先级为 7 （15）。禁止用户中断。
110 = CPU 中断优先级为 6 （14）
101 = CPU 中断优先级为 5 （13）
100 = CPU 中断优先级为 4 （12）
011 = CPU 中断优先级为 3 （11）
010 = CPU 中断优先级为 2 （10）
001 = CPU 中断优先级为 1 （9）
000 = CPU 中断优先级为 0 （8）

注 1： 关于那些没有专用于中断控制功能的剩余位的描述，请参见寄存器 2-1。
2： IPL 位与 IPL3 位 （CORCON<3>）一起构成 CPU 的中断优先级。如果 IPL3 = 1，则括号中的值表示中断优先级。
3： 当 NSTDIS （INTCON1<15>） = 1 时 IPL 状态位为只读。

寄存器 6-2： CORCON：CPU 控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0	R/W-0	U-0	U-0
—	—	—	—	IPL3(2)	PSV(1)	—	—
bit 7							bit 0

图注：	C = 可清零位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 3 **IPL3：**CPU 中断优先级状态位 (2)
1 = CPU 中断优先级大于 7
0 = CPU 中断优先级为 7 或更小

注 1： 关于那些没有专用于中断控制功能的剩余位的描述，请参见寄存器 2-2。
2： IPL3 位与 IPL2:IPL0 位 （SR<7:5>）一起构成 CPU 的中断优先级。

寄存器 6-3： INTCON1：中断控制寄存器 1

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
NSTDIS	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **NSTDIS：**中断嵌套禁止位
1 = 禁止中断嵌套
0 = 允许中断嵌套

bit 14-5 **未实现：**读为 0

bit 4 **MATHERR：**算法错误陷阱状态位
1 = 发生了溢出陷阱
0 = 未发生溢出陷阱

bit 3 **ADDRERR：**地址错误陷阱状态位
1 = 发生了地址错误陷阱
0 = 未发生地址错误陷阱

bit 2 **STKERR：**堆栈错误陷阱状态位
1 = 发生了堆栈错误陷阱
0 = 未发生堆栈错误陷阱

bit 1 **OSCFAIL：**振荡器故障陷阱状态位
1 = 发生了振荡器故障陷阱
0 = 未发生振荡器故障陷阱

bit 0 **未实现：**读为 0

寄存器 6-4： INTCON2：中断控制寄存器 2

R/W-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	INT2EP	INT1EP	INT0EP
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **ALTIVT：**备用中断向量表使能位
1 = 使用备用中断向量表
0 = 使用标准 （默认）向量表

bit 14 **DISI：**DISI 指令状态位
1 = 执行了 DISI 指令
0 = 没有执行 DISI 指令

bit 13-3 **未实现：**读为 0

bit 2 **INT2EP：**外部中断 2 边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

bit 1 **INT1EP：**外部中断 1 边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

bit 0 **INT0EP：**外部中断 0 边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

寄存器 6-5： IFS0：中断标志状态寄存器 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13 **AD1IF：**A/D 转换完成中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 12 **U1TXIF：**UART1 发送器中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 11 **U1RXIF：**UART1 接收器中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 10 **SPI1IF：**SPI1 事件中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 9 **SPF1IF：**SPI1 故障中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 8 **T3IF：**Timer3 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 7 **T2IF：**Timer2 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 6 **OC2IF：**输出比较通道 2 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 5 **IC2IF：**输入捕捉通道 2 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 4 **未实现：**读为 0

bit 3 **T1IF：**Timer1 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 2 **OC1IF：**输出比较通道 1 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 1 **IC1IF：**输入捕捉通道 1 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 0 **INT0IF：**外部中断 0 标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

寄存器 6-6： IFS1：中断标志状态寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IC8IF	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **U2TXIF：**UART2 发送器中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 14 **U2RXIF：**UART2 接收器中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 13 **INT2IF：**外部中断 2 标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 12 **T5IF：**Timer5 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 11 **T4IF：**Timer4 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 10 **OC4IF：**输出比较通道 4 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 9 **OC3IF：**输出比较通道 3 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 8 **未实现：**读为 0

bit 7 **IC8IF：**输入捕捉通道 8 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 6 **IC7IF：**输入捕捉通道 7 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 5 **未实现：**读为 0

bit 4 **INT1IF：**外部中断 1 标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 3 **CNIF：**输入状态变化通知中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 2 **CMIF：**比较器中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 1 **MI2C1IF：**I2C1 主事件中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 0 **SI2C1IF：**I2C1 从事件中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

寄存器 6-7： IFS2：中断标志状态寄存器 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PMPIF	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13 **PMPIF：**并行主端口中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 12 **OC8IF：**输出比较通道 8 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 11 **OC7IF：**输出比较通道 7 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 10 **OC6IF：**输出比较通道 6 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 9 **OC5IF：**输出比较通道 5 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 8 **IC6IF：**输入捕捉通道 6 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 7 **IC5IF：**输入捕捉通道 5 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 6 **IC4IF：**输入捕捉通道 4 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 5 **IC3IF：**输入捕捉通道 3 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 4-2 **未实现：**读为 0

bit 1 **SPI2IF：**SPI2 事件中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 0 **SPF2IF：**SPI2 故障中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

寄存器 6-8： IFS3：中断标志状态寄存器 3

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	RTCIF	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
—	INT4IF	INT3IF	—	—	MI2C2IF	SI2C2IF	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15	**未实现：**读为 0
bit 14	**RTCIF：**实时时钟 / 日历中断标志状态位 1 = 已发生中断请求 0 = 未发生中断请求
bit 13-7	**未实现：**读为 0
bit 6	**INT4IF：**外部中断 4 标志状态位 1 = 已发生中断请求 0 = 未发生中断请求
bit 5	**INT3IF：**外部中断 3 标志状态位 1 = 已发生中断请求 0 = 未产生中断请求
bit 4-3	**未实现：**读为 0
bit 2	**MI2C2IF：**I2C2 主事件中断标志状态位 1 = 已发生中断请求 0 = 未发生中断请求
bit 1	**SI2C2IF：**I2C2 从事件中断标志状态位 1 = 已发生中断请求 0 = 未发生中断请求
bit 0	**未实现：**读为 0

寄存器 6-9： IFS4：中断标志状态寄存器 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	CTMUIF	—	—	—	—	LVDIF
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIF	U2ERIF	U1ERIF	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13 **CTMUIF：**CTMU 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 12-9 **未实现：**读为 0

bit 8 **LVDIF：**低电压检测中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 7-4 **未实现：**读为 0

bit 3 **CRCIF：**CRC 发生器中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 2 **U2ERIF：**UART2 错误中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 1 **U1ERIF：**UART1 错误中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 0 **未实现：**读为 0

寄存器 6-10： IFS5：中断标志状态寄存器 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IC9IF	OC9IF	SPI3IF	SPF3IF	U4TXIF	U4RXIF
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U4ERIF	—	MI2C3IF	SI2C3IF	U3TXIF	U3RXIF	U3ERIF	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13 **IC9IF：**输入捕捉通道 9 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 12 **OC9IF：**输出比较通道 9 中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 11 **SPI3IF：**SPI3 事件中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 10 **SPF3IF：**SPI3 故障中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 9 **U4TXIF：**UART4 发送器中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 8 **U4RXIF：**UART4 接收器中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 7 **U4ERIF：**UART4 错误中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 6 **未实现：**读为 0

bit 5 **MI2C3IF：**I2C3 主事件中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 4 **SI2C3IF：**I2C3 从事件中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 3 **U3TXIF：**UART3 发送器中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 2 **U3RXIF：**UART3 接收器中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 1 **U3ERIF：**UART3 错误中断标志状态位
1 = 已发生中断请求
0 = 未发生中断请求

bit 0 **未实现：**读为 0

寄存器 6-11：　　IEC0：中断允许控制寄存器 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14　**未实现：**读为 0

bit 13　**AD1IE：**A/D 转换完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 12　**U1TXIE：**UART1 发送器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 11　**U1RXIE：**UART1 接收器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 10　**SPI1IE：**SPI1 发送完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 9　**SPF1IE：**SPI1 故障中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 8　**T3IE：**Timer3 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 7　**T2IE：**Timer2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 6　**OC2IE：**输出比较通道 2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 5　**IC2IE：**输入捕捉通道 2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 4　**未实现：**读为 0

bit 3　**T1IE：**Timer1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 2　**OC1IE：**输出比较通道 1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 1　**IC1IE：**输入捕捉通道 1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 0　**INT0IE：**外部中断 0 允许位
1 = 允许中断请求
0 = 禁止中断请求

寄存器 6-12：　IEC1：中断允许控制寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U2TXIE	U2RXIE	INT2IE(1)	T5IE	T4IE	OC4IE	OC3IE	—
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IC8IE	IC7IE	—	INT1IE(1)	CNIE	CMIE	MI2C1IE	SI2C1IE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15　**U2TXIE：**UART2 发送器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 14　**U2RXIE：**UART2 接收器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 13　**INT2IE：**外部中断 2 允许位 (1)
1 = 允许中断请求
0 = 禁止中断请求

bit 12　**T5IE：**Timer5 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 11　**T4IE：**Timer4 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 10　**OC4IE：**输出比较通道 4 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 9　**OC3IE：**输出比较通道 3 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 8　**未实现：**读为 0

bit 7　**IC8IE：**输入捕捉通道 8 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 6　**IC7IE：**输入捕捉通道 7 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 5　**未实现：**读为 0

bit 4　**INT1IE：**外部中断 1 允许位 (1)
1 = 允许中断请求
0 = 禁止中断请求

bit 3　**CNIE：**输入状态变化通知中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 2　**CMIE：**比较器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

注　1： 如果允许外部中断，中断输入必须配置给可用的 RPn 或 RPIn 引脚。更多信息，请参见**第 9.4 节 “外设引脚选择”**。

寄存器 6-12： IEC1：中断允许控制寄存器 1 （续）

bit 1 **MI2C1IE：**I2C1 主事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 0 **SI2C1IE：**I2C1 从事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求

注 1： 如果允许外部中断，中断输入必须配置给可用的 RPn 或 RPIn 引脚。更多信息，请参见**第 9.4 节 " 外设引脚选择 "**。

寄存器 6-13： IEC2：中断允许控制寄存器 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PMPIE	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13 **PMPIE：**并行主端口中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 12 **OC8IE：**输出比较通道 8 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 11 **OC7IE：**输出比较通道 7 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 10 **OC6IE：**输出比较通道 6 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 9 **OC5IE：**输出比较通道 5 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 8 **IC6IE：**输入捕捉通道 6 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 7 **IC5IE：**输入捕捉通道 5 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 6 **IC4IE：**输入捕捉通道 4 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 5 **IC3IE：**输入捕捉通道 3 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 4-2 **未实现：**读为 0

bit 1 **SPI2IE：**SPI2 事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 0 **SPF2IE：**SPI2 故障中断允许位
1 = 允许中断请求
0 = 禁止中断请求

寄存器 6-14： IEC3：中断允许控制寄存器 3

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	RTCIE	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
—	INT4IE(1)	INT3IE(1)	—	—	MI2C2IE	SI2C2IE	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14 **RTCIE：**实时时钟 / 日历中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 13-7 **未实现：**读为 0

bit 6 **INT4IE：**外部中断 4 允许位 (1)
1 = 允许中断请求
0 = 禁止中断请求

bit 5 **INT3IE：**外部中断 3 允许位 (1)
1 = 允许中断请求
0 = 禁止中断请求

bit 4-3 **未实现：**读为 0

bit 2 **MI2C2IE：**I2C2 主事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 1 **SI2C2IE：**I2C2 从事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 0 **未实现：**读为 0

注 1： 如果允许外部中断，中断输入必须配置给可用的 RPn 或 RPIn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

寄存器 6-15： IEC4：中断允许控制寄存器 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	CTMUIE	—	—	—	—	LVDIE
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIE	U2ERIE	U1ERIE	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13	**CTMUIE：**CTMU 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 12-9	**未实现：**读为 0
bit 8	**LVDIE：**低电压检测中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 7-4	**未实现：**读为 0
bit 3	**CRCIE：**CRC 发生器中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 2	**U2ERIE：**UART2 错误中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 1	**U1ERIE：**UART1 错误中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 0	**未实现：**读为 0

寄存器 6-16： IEC5：中断允许控制寄存器 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IC9IE	OC9IE	SPI3IE	SPF3IE	U4TXIE	U4RXIE
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U4ERIE	—	MI2C3IE	SI2C3IE	U3TXIE	U3RXIE	U3ERIE	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13 **IC9IE：**输入捕捉通道 9 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 12 **OC9IE：**输出比较通道 9 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 11 **SPI3IE：**SPI3 事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 10 **SPF3IE：**SPI3 故障中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 9 **U4TXIE：**UART4 发送器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 8 **U4RXIE：**UART4 接收器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 7 **U4ERIE：**UART4 错误中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 6 **未实现：**读为 0

bit 5 **MI2C3IE：**I2C3 主事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 4 **SI2C3IE：**I2C3 从事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 3 **U3TXIE：**UART3 发送器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 2 **U3RXIE：**UART3 接收器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 1 **U3ERIE：**UART3 错误中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 0 **未实现：**读为 0

 超前信息

寄存器 6-17： IPC0：中断优先级控制寄存器 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **T1IP2:T1IP0：**Timer1 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **OC1IP2:OC1IP0：**输出比较通道 1 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **IC1IP2:IC1IP0：**输入捕捉通道 1 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **INT0IP2:INT0IP0：**外部中断 0 优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-18： IPC1：中断优先级控制寄存器 1

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **T2IP2:T2IP0：**Timer2 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **OC2IP2:OC2IP0：**输出比较通道 2 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **IC2IP2:IC2IP0：**输入捕捉通道 2 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

寄存器 6-19： IPC2：中断优先级控制寄存器 2

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **U1RXIP2:U1RXIP0：**UART1 接收器中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **SPI1IP2:SPI1IP0：**SPI1 事件中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **SPF1IP2:SPF1IP0：**SPI1 故障中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **T3IP2:T3IP0：**Timer3 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-20： IPC3：中断优先级控制寄存器 3

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-7 **未实现：**读为 0

bit 6-4 **AD1IP2:AD1IP0：**A/D 转换完成中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **U1TXIP2:U1TXIP0：**UART1 发送器中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-21： IPC4：中断优先级控制寄存器 4

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C1P2	MI2C1P1	MI2C1P0	—	SI2C1P2	SI2C1P1	SI2C1P0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **CNIP2:CNIP0：**输入状态变化通知中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **CMIP2:CMIP0：**比较器中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **MI2C1P2:MI2C1P0：**I2C1 主事件中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **SI2C1P2:SI2C1P0：**I2C1 从事件中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-22： IPC5：中断优先级控制寄存器 5

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC8IP2	IC8IP1	IC8IP0	—	IC7IP2	IC7IP1	IC7IP0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **IC8IP2:IC8IP0：**输入捕捉通道 8 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **IC7IP2:IC7IP0：**输入捕捉通道 7 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7-3 **未实现：**读为 0

bit 2-0 **INT1IP2:INT1IP0：**外部中断 1 优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-23： IPC6：中断优先级控制寄存器 6

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **T4IP2:T4IP0：**Timer4 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **OC4IP2:OC4IP0：**输出比较通道 4 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **OC3IP2:OC3IP0：**输出比较通道 3 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

寄存器 6-24： IPC7：中断优先级控制寄存器 7

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **U2TXIP2:U2TXIP0：**UART2 发送器中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **U2RXIP2:U2RXIP0：**UART2 接收器中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **INT2IP2:INT2IP0：**外部中断 2 优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **T5IP2:T5IP0：**Timer5 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-25： IPC8：中断优先级控制寄存器 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-7 **未实现：**读为 0

bit 6-4 **SPI2IP2:SPI2IP0：**SPI2 事件中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **SPF2IP2:SPF2IP0：**SPI2 故障中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-26： IPC9：中断优先级控制寄存器 9

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **IC5IP2:IC5IP0：**输入捕捉通道 5 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **IC4IP2:IC4IP0：**输入捕捉通道 4 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **IC3IP2:IC3IP0：**输入捕捉通道 3 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

寄存器 6-27： IPC10：中断优先级控制寄存器 10

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC7IP2	OC7IP1	OC7IP0	—	OC6IP2	OC6IP1	OC6IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC5IP2	OC5IP1	OC5IP0	—	IC6IP2	IC6IP1	IC6IP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **OC7IP2:OC7IP0：**输出比较通道 7 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **OC6IP2:OC6IP0：**输出比较通道 6 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **OC5IP2:OC5IP0：**输出比较通道 5 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **IC6IP2:IC6IP0：**输入捕捉通道 6 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-28： IPC11：中断优先级控制寄存器 11

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	PMPIP2	PMPIP1	PMPIP0	—	OC8IP2	OC8IP1	OC8IP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-7 **未实现：**读为 0

bit 6-4 **PMPIP2:PMPIP0：**并行主端口中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **OC8IP2:OC8IP0：**输出比较通道 8 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-29： IPC12：中断优先级控制寄存器 12

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	MI2C2P2	MI2C2P1	MI2C2P0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	SI2C2P2	SI2C2P1	SI2C2P0	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-11 **未实现：**读为 0

bit 10-8 **MI2C2P2:MI2C2P0：**I2C2 主事件中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **SI2C2P2:SI2C2P0：**I2C2 从事件中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

寄存器 6-30： IPC13：中断优先级控制寄存器 13

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT4IP2	INT4IP1	INT4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-11 **未实现：**读为 0

bit 10-8 **INT4IP2:INT4IP0：**外部中断 4 优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **INT3IP2:INT3IP0：**外部中断 3 优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

寄存器 6-31： IPC15：中断优先级控制寄存器 15

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-11 **未实现：**读为 0

bit 10-8 **RTCIP2:RTCIP0：**实时时钟 / 日历中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7-0 **未实现：**读为 0

寄存器 6-32： IPC16：中断优先级控制寄存器 16

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **CRCIP2:CRCIP0：**CRC 发生器错误中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **U2ERIP2:U2ERIP0：**UART2 错误中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **U1ERIP2:U1ERIP0：**UART1 错误中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

寄存器 6-33： IPC18：中断优先级控制寄存器 18

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-3 **未实现：**读为 0

bit 2-0 **LVDIP2:LVDIP0：**低电压检测中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-34： IPC19：中断优先级控制寄存器 19

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-7 **未实现：**读为 0

bit 6-4 **CTMUIP2:CTMUIP0：**CTMU 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

寄存器 6-35： IPC20：中断优先级控制寄存器 20

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **U3TXIP2:U3TXIP0：**UART3 发送器中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **U3RXIP2:U3RXIP0：**UART3 接收器中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **U3ERIP2:U3ERIP0：**UART3 错误中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

寄存器 6-36： IPC21：中断优先级控制寄存器 21

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U4ERIP2	U4ERIP1	U4ERIP0	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C3P2	MI2C3P1	MI2C3P0	—	SI2C3P2	SI2C3P1	SI2C3P0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **U4ERIP2:U4ERIP0：**UART4 错误中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11-7 **未实现：**读为 0

bit 6-4 **MI2C3P2:MI2C3P0：**I2C3 主事件中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **SI2C3P2:SI2C3P0：**I2C3 从事件中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-37： IPC22：中断优先级控制寄存器 22

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI3IP2	SPI3IP1	SPI3IP0	—	SPF3IP2	SPF3IP1	SPF3IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **SPI3IP2:SP3IP0：**SPI3 事件中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **SPF3IP2:SPF3IP0：**SPI3 故障中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **U4TXIP2:U4TXIP0：**UART4 发送器中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **U4RXIP2:U4RXIP0：**UART4 接收器中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

寄存器 6-38： IPC23：中断优先级控制寄存器 23

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC9IP2	IC9IP1	IC9IP0	—	OC9IP2	OC9IP1	OC9IP0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-7 **未实现：**读为 0

bit 6-4 **IC9IP2:IC9IP0：**输入捕捉通道 9 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **OC9IP2:OC9IP0：**输出比较通道 9 中断优先级位
111 = 中断优先级为 7 （最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

6.4 中断设置过程

6.4.1 初始化

按以下步骤配置中断源：

1. 如果不需要嵌套中断，则将 NSTDIS 控制位（INTCON1<15>）置 1。
2. 通过写入相应 IPCx 寄存器中的控制位来为中断源选择用户分配的优先级。优先级将取决于特定的应用和中断源的类型。如果不需要多个优先级，可以将所有允许中断源的 IPCx 寄存器控制位编程为相同的非零值。

注：	在器件复位时，IPCx 寄存器被初始化，为所有用户中断源分配优先级 4。

3. 将相应 IFSx 寄存器中与外设相关的中断标志状态位清零。
4. 通过将相应 IECx 寄存器中与中断源相关的中断允许控制位置 1 来允许中断源。

6.4.2 中断服务程序

如何声明中断服务程序（ISR）以及怎样使用正确的向量地址初始化 IVT，将取决于编程语言（即 C 语言或汇编语言）和用于开发应用程序的语言开发工具包。一般情况下，用户必须将相应 IFSx 寄存器中与 ISR 处理的中断源相对应的中断标志清零。否则，在退出 ISR 后会立即再次进入 ISR。如果 ISR 用汇编语言编写，则必须使用 `RETFIE` 指令结束 ISR，以便将保存的 PC 值、SRL 值和原先的 CPU 优先级弹出堆栈。

6.4.3 陷阱服务程序

陷阱服务程序（Trap Service Routine， TSR）的编码方式类似于 ISR，只是必须将 INTCON1 寄存器中相应的陷阱状态标志清零，以避免重新进入 TSR。

6.4.4 禁止中断

可以通过以下步骤禁止所有用户中断：

1. 使用 `PUSH` 指令将当前 SR 值压入软件堆栈。
2. 通过将值 OEh 与 SRL 进行逻辑或运算来强制把 CPU 的优先级设置为 7。

要允许用户中断，可以使用 `POP` 指令恢复先前的 SR 值。

注意只能禁止优先级小于或等于 7 的用户中断。不能禁止陷阱源（优先级为 8-15）。

使用 `DISI` 指令可以方便地将优先级为 1-6 的中断禁止一段固定的时间。 `DISI` 指令不能禁止优先级为 7 的中断源。

7.0 振荡器配置

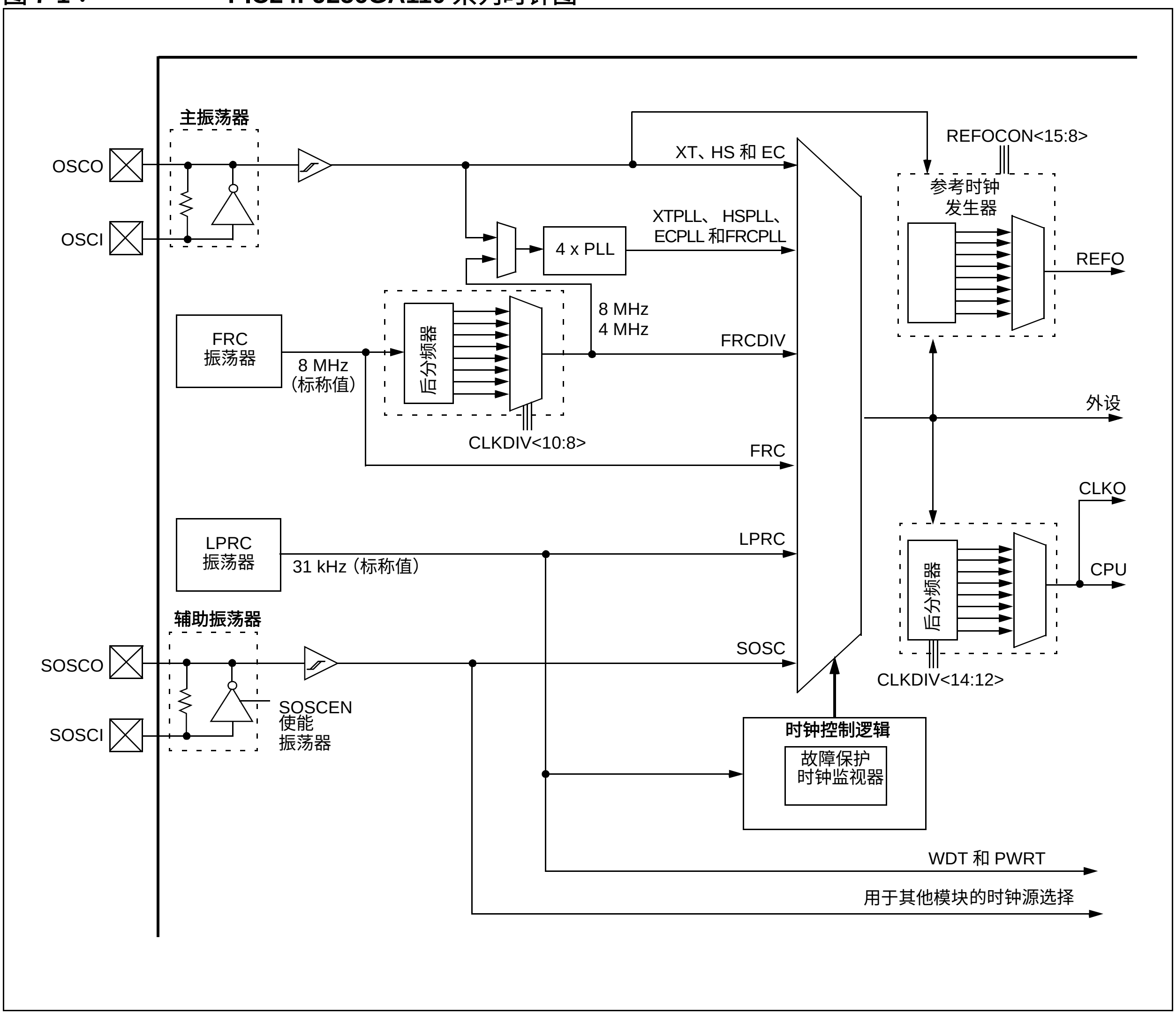
> **注：** 本数据手册总结了 PIC24F 系列器件的功能。但是部包括全部参考资料。更多信息，请参见《PIC24F 系列参考手册》中的 **“ 第 6 章 振荡器 ”**（DS39700A_CN）。

PIC24FJ256GA110 系列器件的振荡器系统具有以下特性：

- 共有四个外部和内部振荡器可选作为时钟源，提供 11 种不同的时钟模式
- 片上 4 倍频 PLL 可以提高所选的内部和外部振荡器源的内部工作频率
- 可采用软件控制在多个时钟源之间切换
- 由软件控制的后分频器对 CPU 的时钟进行选择以节省系统的功耗
- 检测时钟故障和允许安全恢复或关闭应用的故障保护时钟监视器 （Fail-Safe Clock Monitor， FSCM）
- 用于同步外部硬件的可独立配置的系统时钟输出。

图 7-1 所示为振荡器系统的简化框图。

图 7-1： PIC24FJ256GA110 系列时钟图



7.1 CPU 时钟机制

可使用四个时钟源中的一个提供系统时钟：

- OSCI 和 OSCO 引脚上的主振荡器 （POSC）
- SOSCI 和 SOSCO 引脚上的辅助振荡器 （SOSC）
- 内部快速 RC （FRC）振荡器
- 内部低功耗 RC （LPRC）振荡器

主振荡器和 FRC 源可以选择使用内部 4x PLL。可编程的时钟分频器可选择性地降低 FRC 时钟源的频率。选择的时钟源提供处理器和外设的时钟。

将处理器时钟源两分频可以产生内部指令周期时钟 FCY。在本文档中，指令周期时钟也可用 FOSC/2 表示。在 OSCO I/O 引脚上可以提供内部指令周期时钟 FOSC/2，用于主振荡器的某些工作模式。

7.2 发生 POR 时的初始配置

通过使用配置位设置选择器件发生上电复位事件时使用的振荡器源（和工作模式）。振荡器配置位设置位于程序存储器中的配置寄存器中（更多详细信息请参见 **第 24.1 节 " 配置位 "**）。主振荡器配置位 POSCMD1:POSCMD0 （配置字 2<1:0>）和初始振荡器选择配置位 FNOSC2:FNOSC0 （配置字 2<10:8>）选择上电复位时使用的振荡器源。带有后分频器的 FRC主振荡器 （FRCDIV）是默认 （未编程）的选择。对这些位所在的位置进行编程可以选择辅助振荡器或其中一个内部振荡器。

配置位允许用户在多个时钟模式中进行选择，如表 7-1 所示。

7.2.1 时钟切换模式配置位

FCKSM 配置位（配置字 2<7:6>）用于同时配置器件时钟切换和故障保护时钟监视器 （FSCM） 。只有将 FCKSM1 编程为 `0` 才可以使能时钟切换功能。只有当 FCKSM1:FCKSM0 被编程为 `00` 才可以使能 FSCM。

表 7-1： 用于时钟选择的配置位值

振荡器模式	振荡器源	POSCMD1: POSCMD0	FNOSC2: FNOSC0	注
带有后分频器的快速 RC 振荡器（FRCDIV）	内部	11	111	**1, 2**
（保留）	内部	xx	110	**1**
低功耗 RC 振荡器 （LPRC）	内部	11	101	**1**
辅助 （Timer1）振荡器 （SOSC）	辅助	11	100	**1**
带有 PLL 模块的主振荡器 （XT）（XTPLL）	主	01	011	
带有 PLL 模块的主振荡器 （EC）（ECPLL）	主	00	011	
主振荡器 （HS）	主	10	010	
主振荡器 （XT）	主	01	010	
主振荡器 （EC）	主	00	010	
带有 PLL 模块的快速 RC 振荡器（FRCPLL）	内部	11	001	**1**
快速 RC 振荡器 （FRC）	内部	11	000	1

注 1： OSCO 引脚功能由 OSCIOFCN 配置位决定。
2： 对于未编程 （已擦除）器件，这是默认的振荡器模式。

7.3 控制寄存器

振荡器的操作由三个特殊功能寄存器控制：

- OSCCON
- CLKDIV
- OSCTUN

OSCCON 寄存器 （寄存器 7-1）是控制振荡器的主要寄存器。该寄存器控制时钟源的切换并允许监视时钟源。

CLKDIV 寄存器 （寄存器 7-2）控制打盹模式的相关特性，以及 FRC 振荡器的后分频器。

OSCTUN寄存器（寄存器7-3）允许用户在大约±12%的范围内精确地调整 FRC 振荡器。递增或递减每个位可将 FRC 振荡器的出厂校准频率更改固定的值。

寄存器 7-1： OSCCON：振荡器控制寄存器

U-0	R-0	R-0	R-0	U-0	R/W-x(1)	R/W-x(1)	R/W-x(1)
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15							bit 8

R/SO-0	R/W-0	R-0(3)	U-0	R/CO-0	R/W-0	R/W-0	R/W-0
CLKLOCK	IOLOCK(2)	LOCK	—	CF	POSCEN	SOSCEN	OSWEN
bit 7							bit 0

图注：	CO = 只清零位	SO = 只置 1 位	
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **未实现：**读为 0

bit 14-12 **COSC2:COSC0：**当前振荡器选择位
111 = 带有后分频器的快速 RC 振荡器 （FRCDIV）
110 = 保留
101 = 低功耗 RC 振荡器 （LPRC）
100 = 辅助振荡器 （SOSC）
011 = 带有 PLL 模块的主振荡器 （XTPLL、 HSPLL 和 ECPLL）
010 = 主振荡器 （XT、 HS 和 EC）
001 = 带有后分频器和 PLL 模块的快速 RC 振荡器 （FRCPLL）
000 = 快速 RC 振荡器 （FRC）

bit 11 **未实现：**读为 0

bit 10-8 **NOSC2:NOSC0：**新振荡器选择位 (1)
111 = 带有后分频器的快速 RC 振荡器 （FRCDIV）
110 = 保留
101 = 低功耗 RC 振荡器 （LPRC）
100 = 辅助振荡器 （SOSC）
011 = 带有 PLL 模块的主振荡器 （XTPLL、 HSPLL 和 ECPLL）
010 = 主振荡器 （XT、 HS 和 EC）
001 = 带有后分频器和 PLL 模块的快速 RC 振荡器 （FRCPLL）
000 = 快速 RC 振荡器 （FRC）

注 1： 这些位的复位值由 FNOSC 配置位决定。
2： 只能在执行解锁序列后更改 IOLOCK 位的状态。另外，如果 IOL1WAY 配置位为 1，一旦 IOLOCK 位置 1，就不能清零。
3： 在进行有效的时钟切换或选择非 PLL 时钟模式时，也复位为 0。

寄存器 7-1：　OSCCON：振荡器控制寄存器　（续）

bit 7　**CLKLOCK：**时钟选择锁定使能位
如果使能 FSCM （FCKSM1 = 1）：
1 = 锁定时钟和 PLL 选择
0 = 不锁定时钟和 PLL 选择，可通过将 OSWEN 位置 1 来修改时钟和 PLL 选择
如果禁止 FSCM （FCKSM1 = 0）：
不锁定时钟和 PLL 选择，可通过将 OSWEN 位置 1 来修改时钟和 PLL 选择。

bit 6　**IOLOCK：**I/O 锁定使能位 [(2)]
1 = I/O 锁定有效
0 = I/O 锁定无效

bit 5　**LOCK：**PLL 锁定状态位 [(3)]
1 = PLL 模块处于锁定状态或 PLL 模块启动定时器到时
0 = PLL 模块不处于锁定状态， PLL 启动定时器正在运行或 PLL 被禁止

bit 4　**未实现：**读为 0

bit 3　**CF：**时钟故障检测位
1 = FSCM 检测到一个时钟故障
0 = 未检测到时钟故障

bit 2　**POSCEN：**主振荡器休眠使能位
1 = 在休眠模式下，主振荡器继续工作
0 = 在休眠模式下，主振荡器停止工作

bit 1　**SOSCEN：**32 kHz 辅助振荡器 （SOSC）使能位
1 = 使能辅助振荡器
0 = 禁止辅助振荡器

bit 0　**OSWEN：**振荡器切换使能位
1 = 振荡器切换到由 NOSC2:NOSC0 位指定的时钟源
0 = 完成振荡器切换

注　1： 这些位的复位值由 FNOSC 配置位决定。
2： 只能在执行解锁序列后更改 IOLOCK 位的状态。另外，如果 IOL1WAY 配置位为 1，一旦 IOLOCK 位置 1，就不能清零。
3： 在进行有效的时钟切换或选择非 PLL 时钟模式时，也复位为 0。

寄存器 7-2： CLKDIV：时钟分频器寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
ROI	DOZE2	DOZE1	DOZE0	DOZEN(1)	RCDIV2	RCDIV1	RCDIV0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **ROI：**中断时恢复位
1 = 中断将 DOZEN 位清零并将 CPU 和外设的时钟比复位为 1:1
0 = 中断不影响 DOZEN 位

bit 14-12 **DOZE2:DOZE0：**CPU 和外设的时钟比选择位
111 = 1:128
110 = 1:64
101 = 1:32
100 = 1:16
011 = 1:8
010 = 1:4
001 = 1:2
000 = 1:1

bit 11 **DOZEN：**打盹使能位 (1)
1 = DOZE2:DOZE0 位指定 CPU 和外设的时钟比
0 = 将 CPU 和外设的时钟比设置为 1:1

bit 10-8 **RCDIV2:RCDIV0：**FRC 后分频比选择位
111 = 31.25 kHz （256 分频）
110 = 125 kHz （64 分频）
101 = 250 kHz （32 分频）
100 = 500 kHz （16 分频）
011 = 1 MHz （8 分频）
010 = 2 MHz （4 分频）
001 = 4 MHz （2 分频）
000 = 8 MHz （1 分频）

bit 7-0 **未实现：**读为 0

注 1： 当 ROI 位置 1 并发生一个中断时，该位会自动清零。

寄存器 7-3： OSCTUN：FRC 振荡器调节寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN5(1)	TUN4(1)	TUN3(1)	TUN2(1)	TUN1(1)	TUN0(1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-6 **未实现：**读为 0

bit 5-0 **TUN5:TUN0：**FRC 振荡器调节位

011111 = 最大频率偏差
011110 =
•
•
•
000001 =
000000 = 中心频率，振荡器运行在经过出厂前校准的频率上
111111 =
•
•
•
100001 =
100000 = 最小频率偏差

注 1： TUN5:TUN0 的递增值或递减值不会在 FRC 调节范围内同步更改 FRC 频率，且频率的变化也可能不是单调的。

7.4 时钟切换工作原理

在极少的限制下，应用可在软件控制下随时在四个时钟源（POSC、SOSC、FRC 和 LPRC）间自由切换。为限制该灵活性可能产生的负面影响， PIC24F 器件的时钟切换过程带有安全锁定。

> **注：** 主振荡器模式有三种不同的子模式 （XT、HS 和 EC），这三个子模式由 POSCMDx 配置位决定。在应用中可以用软件实现从主振荡器模式切换到其他模式，或从其他模式切换到主振荡器模式，但不能在不对器件进行再编程的情况下在主振荡器模式的不同子模式之间进行切换。

7.4.1 使能时钟切换

要使能时钟切换，CW 2 中的 FCKSM1 配置位必须编程为 0。（更多详细信息，请参见**第 24.1 节 " 配置位 "**）。如果没有将 FCKSM1 配置位编程为 1，则时钟切换功能和故障保护时钟监视器功能都将被禁止。这是默认设置。

在时钟切换被禁止的情况下， NOSCx 控制位（OSCCON<10:8>）不控制时钟选择。但是， COSCx 位 （OSCCON<14:12>）将会反映 FNOSCx 配置位选择的时钟源。

禁止时钟切换时，OSWEN 控制位（OSCCON<0>）无效。它总是保持为 0。

 超前信息

7.4.2 振荡器切换序列

执行时钟切换至少需要下列基本序列：

1. 如果需要，读 COSCx 位 （OSCCON<14:12>）确定当前振荡器源。
2. 执行解锁序列以允许写入 OSCCON 寄存器的高字节。
3. 将新振荡器源的相关值写入 NOSCx 位（OSCCON<10:8>）。
4. 执行解锁序列以允许写入 OSCCON 寄存器的低字节。
5. 将 OSWEN 位置 1 来启动振荡器切换。

一旦基本序列完成，系统时钟硬件将自动进行如下响应：

1. 时钟切换硬件将NOSCx控制位的新值和COSCx状态位做比较。如果相等，时钟切换为冗余操作。在这种情况下，OSWEN 位被自动清零且时钟切换被中止。
2. 如果启动了有效时钟切换，则LOCK（OSCCON<5>）和 CF （OSCCON<3>）状态位被清零。
3. 如果新振荡器现在不运行，硬件会将其开启。如果必须打开晶振，硬件将等待直至 OST 超时。如果新振荡源正在使用 PLL，硬件将等待直到检测到 PLL 锁定 （LOCK = 1）。
4. 硬件会等待新时钟源的 10 个时钟周期，然后执行时钟切换。
5. 硬件清零 OSWEN 位表示时钟切换成功。此外，NOSCx 位的值被传送到 COSCx 位中。
6. 此时旧时钟源被关闭，但 LPRC（如果使能 WDT 或 FSCM）或 SOSC （如果 SOSCEN 位保持置 1 状态）除外。

> **注 1：** 在整个时钟切换过程中，处理器将继续执行代码。对时序要求高的代码不应在此时执行。
>
> **2：** 不允许直接在使能 PLL 的任何主振荡器模式和 FRCPLL 模式之间进行时钟切换。反之亦然。在这些情况下，应用必须首先切换到 FRC 模式将其作为两个 PLL 模式之间的过渡时钟源。

推荐的时钟切换的代码序列通常包括：

1. 在 OSCCON 寄存器解锁和写序列过程中禁止中断。
2. 通过两条连续的指令将 78h 和 9Ah 写入OSCCON<15:8>来执行OSCCON高字节的解锁序列。
3. 执行解锁序列之后，立即使用指令将新振荡器源写入 NOSCx 位。
4. 通过两条连续的指令将 46h 和 57h 写入OSCCON<7:0> 来执行 OSCCON 低字节的解锁序列。
5. 执行解锁序列之后，立即将指令中的 OSWEN 位置 1。
6. 继续执行对时钟要求不高的代码 （可选）。
7. 调用适当时间的软件延时 （周期计数），以允许选定的振荡器和 （或） PLL 启动并稳定下来。
8. 检查OSWEN位是否为0。如果是，则切换成功。如果 OSWEN 位仍为置 1 状态，则检查 LOCK 位以确定故障原因。

解锁 OSCCON 寄存器并启动时钟切换的核心序列如例 7-1 所示。

例 7-1： 时钟切换的基本代码序列

```
;Place the new oscillator selection in W0
;OSCCONH (high byte) Unlock Sequence
MOV         #OSCCONH, w1
MOV         #0x78, w2
MOV         #0x9A, w3
MOV.b       w2, [w1]
MOV.b       w3, [w1]
;Set new oscillator selection
MOV.b       WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV         #OSCCONL, w1
MOV         #0x46, w2
MOV         #0x57, w3
MOV.b       w2, [w1]
MOV.b       w3, [w1]
;Start oscillator switch operation
BSET        OSCCON,#0
```

7.5 参考时钟输出

除了 CLKO 输出（Fosc/2）在某一振荡器模式中可用之外，PIC24FJ256GA110 系列器件中的器件时钟还可以被配置为向端口引脚提供参考时钟输出。该功能适用于全部振荡器配置并允许用户选择更大范围的时钟分频比以驱动应用程序中的外部器件。

该参考时钟输出由REFOCON 寄存器（寄存器 7-4）控制。将 ROEN 位（REFOCON<15>）置 1 使 REFO 引脚上的时钟信号可用。RODIV 位（REFOCON<11:8>）可以在 16 个不同的时钟分频比选项中进行选择。

ROSSLP 和 ROSEL 位（REFOCON<13:12>）控制休眠模式下参考输出的可用性。ROSEL 位确定 OSC1 和 OSC2 上的振荡器或当前系统时钟源是否用于参考时钟输出。ROSSLP 位确定当器件处于休眠模式下时，参考源在 REFO 上是否可用。

要在休眠模式下使用参考时钟输出，必须将ROSSLP和ROSEL 位置 1。器件时钟必须被配置为 EC、HS 或 XT 几种主模式之一；否则如果未将 POSCEN 位置 1，则 OSC1 和 OSC2 上的振荡器将在器件进入休眠模式时掉电。将 ROSEL 位清零可以允许在时钟切换期间参考输出频率随系统时钟频率的变化而变化。

寄存器 7-4： REFOCON：参考振荡器控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **ROEN：**参考振荡器输出使能位
1 = 在 REFO 引脚上使能参考振荡器
0 = 禁止参考振荡器

bit 14 **未实现：**读为 0

bit 13 **ROSSLP：**休眠模式下参考振荡器输出停止位
1 = 休眠模式下参考振荡器继续运行
0 = 休眠模式下参考振荡器停止运行

bit 12 **ROSEL：**参考振荡器源选择位
1 = 主振荡器用作基本时钟。注意，必须使用 FOSC2:FOSC0 位使能晶体振荡器；晶体振荡器可在休眠模式下继续工作。
0 = 系统时钟用作基本时钟；基本时钟会反映器件的任何时钟切换。

寄存器 7-4： REFOCON：参考振荡器控制寄存器 （续）

bit 11-8 **RODIV3:RODIV0：**参考振荡器分频比选择位
1111 = 基本时钟的 32,768 分频比
1110 = 基本时钟的 16,384 分频比
1101 = 基本时钟的 8,192 分频比
1100 = 基本时钟的 4,096 分频比
1011 = 基本时钟的 2,048 分频比
1010 = 基本时钟的 1,024 分频比
1001 = 基本时钟的 512 分频比
1000 = 基本时钟的 256 分频比
0111 = 基本时钟的 128 分频比
0110 = 基本时钟的 64 分频比
0101 = 基本时钟的 32 分频比
0100 = 基本时钟的 16 分频比
0011 = 基本时钟的 8 分频比
0010 = 基本时钟的 4 分频比
0001 = 基本时钟的 2 分频比
0000 = 基本时钟

bit 7-0 **未实现：**读为 0

注：

8.0 节能功能

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》中的**“第 10 章 节能功能”**（DS39698A_CN）。

PIC24FJ256GA110 系列器件提供了通过有选择地管理 CPU 和外设的时钟源来管理功耗的功能。一般而言，较低的时钟频率和减少时钟源驱动电路的数目会使功耗降低。所有 PIC24F 器件通过以下四种方法管理功耗：

- 时钟频率
- 基于指令的休眠和空闲模式
- 软件控制的打盹模式
- 通过软件有选择地进行外设控制

可以组合使用这些方法，从而在保证应用关键功能（如对于时序要求高的通信）的情况下有选择地调节应用的功耗。

8.1 时钟频率和时钟切换

PIC24F 器件提供的时钟频率范围较大，用户可根据应用需要进行选择。如果未锁定系统时钟配置，用户只需更改 NOSC 配置位即可选择低功耗或高精度振荡器。在工作期间更改系统时钟的过程以及该过程的限制，已在**第 7.0 节 “振荡器配置”**中进行了更详细地讨论。

8.2 基于指令的节能模式

PIC24F 器件有两种特殊的节能模式，通过执行特殊的 PWRSAV 指令可以进入这两种模式。休眠模式下时钟停止运行并停止所有代码执行；空闲模式下 CPU 停止工作，代码停止执行，但是允许外设模块继续工作。例 8-1 中所示为 PWRSAV 指令的汇编语法。

允许的中断、 WDT 超时或器件复位会导致器件从休眠和空闲模式退出。器件退出这两种模式的过程称为 “唤醒 ”。

注：	SLEEP_MODE 和 IDLE_MODE 是所选器件的汇编器头文件中定义的常数。

8.2.1 休眠模式

休眠模式有以下三种特性：

- 系统时钟源关闭。如果使用了片上振荡器，它也将关闭。
- 如果没有 I/O 引脚拉电流，则器件电流消耗将降至最低。
- 由于禁止了系统时钟源，所以故障保护时钟监视器在休眠模式下不工作。
- 如果 WDT 被使能， LPRC 时钟将继续在休眠模式下运行。
- 若使能了 WDT，它将在进入休眠模式之前自动清零。
- 某些器件功能或外设可在休眠模式下继续工作。这包括 I/O 端口上的输入变化通知功能或使用外部时钟输入的外设等。任何需要系统时钟源工作的外设都会在休眠模式下禁止。

当发生以下任何事件时，器件将从休眠模式唤醒：

- 任何被单独允许的中断
- 任何形式的器件复位
- WDT 超时

从休眠模式下唤醒时，处理器将使用在进入休眠模式时的有效时钟源重新启动。

例 8-1： PWRSAV **指令语法**

```
PWRSAV    #SLEEP_MODE       ; Put the device into SLEEP mode
PWRSAV    #IDLE_MODE        ; Put the device into IDLE mode
```

8.2.2 空闲模式

空闲模式有以下几种特性：

- CPU 将停止执行指令。
- WDT 自动清零。
- 系统时钟源保持活动状态。默认情况下，所有外设模块将使用系统时钟源继续正常工作，但也可被有选择地禁止 （见**第 8.4 节 “ 有选择的外设模块控制 ”**）。
- 如果 WDT 或 FSCM 被使能， LPRC 也将保持活动状态。

当发生以下任何事件时，器件将从空闲模式唤醒：

- 产生被单独允许的中断。
- 任何器件复位。
- WDT 超时。

从空闲模式唤醒时，重新为 CPU 提供时钟，且立即从 PWRSAV 指令之后的下一条指令或 ISR 中的第一条指令开始执行指令。

8.2.3 在节能指令执行期间的中断

在 PWRSAV 指令执行期间发生的中断都将延迟到进入休眠或空闲模式后才产生。并导致器件从休眠或空闲模式下唤醒。

8.3 打盹模式

通常，更改时钟速度和进入某种节能模式是降低功耗的首选策略。然而，有些情况下不可行。例如，某些应用可能必须保持不间断的同步通信，即使在它不执行任何其他操作时也不例外。降低系统时钟速度可能会导致通信错误，而使用节能模式就可能完全停止通信。

打盹模式是另一种简单有效的节能方法，它可以在器件仍然执行代码的情况下降低功耗。在此模式下，继续以相同的时钟源和相同的速度驱动系统时钟。外设模块时钟速度保持不变，但 CPU 时钟的速度降低了。保持两个时钟域同步，以允许外设在 CPU 以较慢的速率执行代码时访问 SFR。

通过将 DOZEN 位（CLKDIV<11>）置 1 使能打盹模式。外设和内核时钟速度之比由 DOZE2:DOZE0 位（CLKDIV<14:12>）决定。有八种可能的配置，从 1:1 到 1:256，其中 1:1 是默认设置。

还可使用打盹模式在事件驱动应用中有选择地降低功耗。这样就可以实现不间断地运行对时序要求高的功能（如同步通信），而 CPU 保持空闲，等待事件调用中断程序。通过将 ROI 位 （CLKDIV<15>）置 1，可以使器件在产生中断时自动返回到全速 CPU 工作模式。默认情况下，中断事件对打盹模式工作没有影响。

8.4 有选择的外设模块控制

空闲和打盹模式允许用户通过降低 CPU 时钟速度或停止 CPU 时钟大幅降低功耗。即使如此，外设模块仍然使用时钟源并因此耗能。可能在有些情况下应用需要这些模式未提供的功能，比如要将功率资源分配给CPU处理工作，而外设的功耗最低。

PIC24F 器件通过允许有选择地禁止外设模块，从而降低或消除它们的功耗，以此满足上述需求。可通过两个控制位完成此操作：

- 外设使能位，通常称为 “ XXXEN ”，位于模块主控制 SFR 中。
- 外设模块禁止 （Peripheral Module Disable，PMD）位，通常称为 “ XXXMD ”，位于某个 PMD 控制寄存器中。

这两个位在使能或禁止相关模块时具有相似的功能。将某个模块的 PMD 位置 1 会禁止该模块的所有时钟源，从而将其功耗降至绝对最小值。在此状态下，与此外设相关的控制和状态寄存器也会被禁止，所以无法写这些寄存器且读取值无效。很多外设模块都有对应的 PMD 位。

通过清零某个模块的 XXXEN 位将会禁止其功能，但是仍然允许对其寄存器进行读写操作。这样做会降低功耗，但是效果没有将 PMD 位置 1 那么大。大多数外设模块都有一个使能位；但捕捉、比较和 RTCC 模块除外。

要节省更多的功耗，也可在器件进入空闲模式时有选择地禁止外设模块。使用通用名格式为 “ XXXIDL ” 的控制位可以执行此操作。默认情况下，可以在空闲模式下工作的所有模块都可以执行此操作。使用 “ 在空闲模式下禁止 ” 功能可以进一步降低空闲模式下的功耗，从而可将节省下来的功耗用于应用中的关键部分。

 超前信息

9.0 I/O 端口

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的 **" 第 12 章 具有外设引脚选择（PPS）功能的 I/O 引脚 "**（DS39711A_CN）。

所有器件引脚（除了 V_{DD}、V_{SS}、$\overline{MCLR}$ 和 OSCI/CLKI 以外）均由外设和并行 I/O 端口共用。所有 I/O 输入端口都为施密特触发器输入，以便增强抗干扰性。

9.1 并行 I/O （PIO）端口

通常，与某个外设共用一个引脚的并行 I/O （PIO）端口总是服从于该外设。外设的输出缓冲器数据和控制信号提供给一对多路开关。这对多路开关用于选择 I/O 引脚的输出数据和控制信号是用于外设还是相应的端口。该逻辑电路同时会阻止 " 环回进入（loop through）"，即一个端口的数字输出可以驱动共用同一个引脚的外设输入。图 9-1 中显示出端口是如何与其他外设复用的，以及对应的 I/O 引脚。

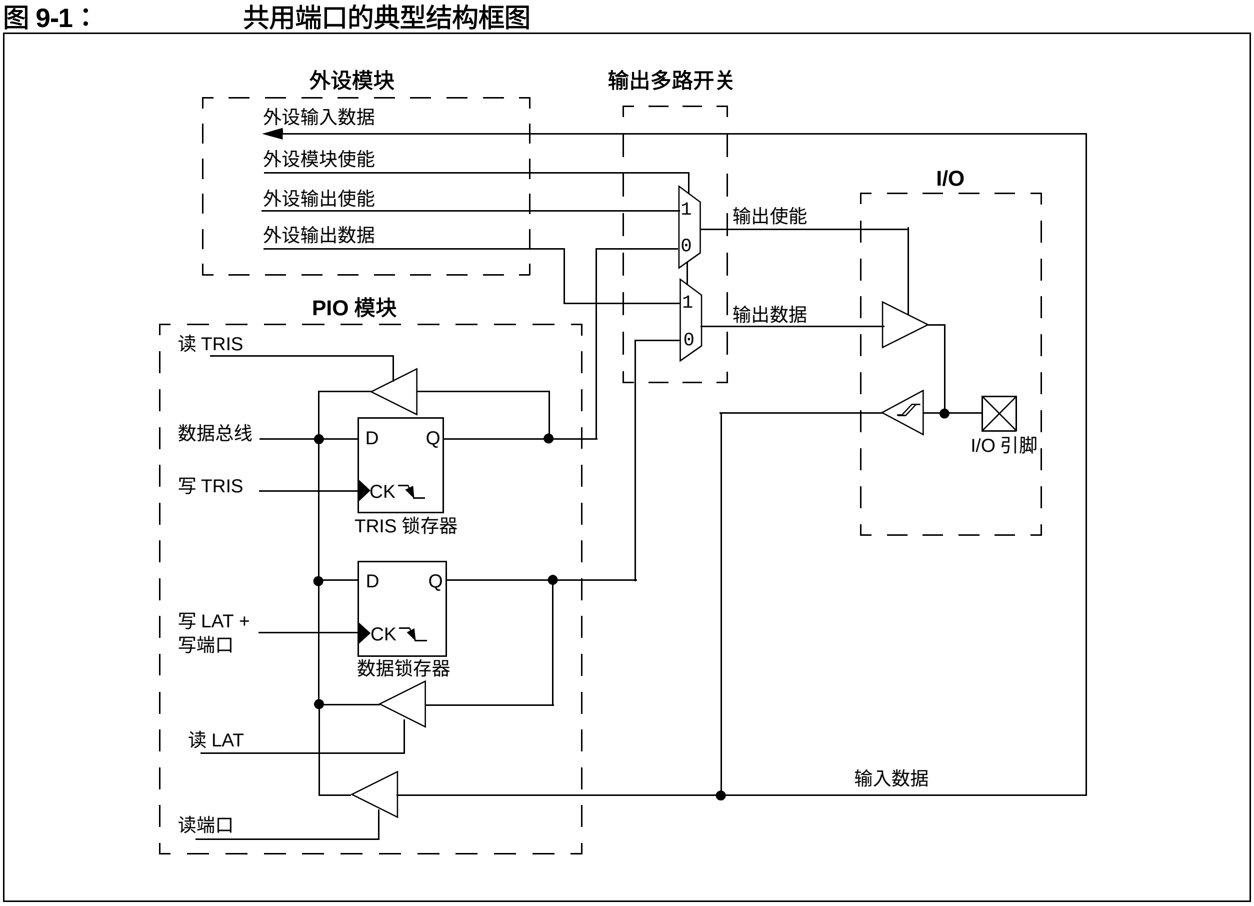
当使能某外设并驱动与其相对应的引脚时，将禁止此引脚的通用输出功能。可以读该 I/O 引脚，但并行端口位的输出驱动器将被禁止。若使能某外设但没有驱动引脚时，则该引脚可由一个端口驱动。

所有端口引脚都有三个寄存器，这些寄存器与端口引脚作为数字 I/O 时的工作直接相关。数据方向寄存器（TRISx）决定引脚是输入引脚还是输出引脚。如果数据方向位为 " 1 "，则为输入引脚。复位以后，所有端口引脚被定义为输入引脚。读输出锁存寄存器（LATx）时，读到的是锁存器中的值；写锁存器时，写入的是锁存器。但读取端口 （PORTx）时，读到的是端口引脚的值；而写入端口引脚时，写入的是相应的锁存器。

对于某个特定器件，无效的位及其相关的数据和控制寄存器都将被禁止。这意味着相应的 LATx 和 TRISx 寄存器以及该端口引脚将读为零。

当一个定义为只用作输入的引脚与另一个外设或功能复用时，由于没有其他竞争的输出源，它将被认为是一个专用端口。

图 9-1： 共用端口的典型结构框图



9.1.1 漏极开路配置

除 PORT、 LAT 和 TRIS 寄存器用于数据控制外，每个端口引脚也可被单独地配置为数字输出或漏极开路输出。这是由与每个端口相对应的漏极开路控制寄存器 ODCx 控制的。将其中的任何位置 1 即可将相应的引脚配置为漏极开路输出。

这种开漏特性允许通过使用外部上拉电阻在任何只能用作数字功能的引脚上产生高于 VDD （如 5V）的输出电平。允许的最大开漏电压与最大 VIH 规范相同。

9.2 配置模拟端口引脚

AD1PCFG 和 TRIS 寄存器用于控制 A/D 端口引脚的操作。将端口引脚设置为模拟输入需要将相应的 TRIS 位置 1。若将 TRIS 位清零 （输出），则该引脚的数字输出电平 （VOH 或 VOL）将被转换。

当对 PORT 寄存器进行读操作时，所有配置为模拟输入通道的引脚将被读为零 （低电平）。

配置为数字输入的引脚将不会对模拟输入进行转换。任何定义为数字输入的引脚 （包括 ANx 引脚）上的模拟电平可能导致输入缓冲器消耗的电流超过器件规范限定值。

9.2.1 I/O 端口写 / 读时序

在改变端口方向或对端口执行写操作，与对同一端口执行读操作之间需要间隔一个指令周期。通常，在两者之间插一条 NOP 指令。

9.3 输入状态变化通知

I/O 端口的输入状态变化通知功能允许 PIC24FJ256GA110 系列器件在选定输入引脚的状态变化时，向处理器发出中断请求。当禁止时钟时，该特性可在休眠模式下检测到输入状态改变事件。取决于器件的引脚数，最多可以选择（允许） 81 个外部信号在输入状态发生变化时产生中断请求。

CNEN1 至 CNEN6 寄存器包含每个 CN 输入引脚的中断允许控制位。将其中任一位置 1 将允许相应引脚的 CN 中断。

每个 CN 引脚都与一个弱上拉和一个弱下拉相连接。弱上拉电路为该引脚提供拉电流，弱下拉电路为该引脚提供灌电流。当连接了按钮或键盘设备时，不再需要使用外部电阻。使用包含弱上拉使能位的 CNPU1 至 CNPU6 寄存器和包含弱下拉使能位的 CNPD1 至 CNPD6 寄存器可分别使能上拉电路和下拉电路。每个 CN 引脚都独立控制其上拉位和下拉位。将控制位置 1 可使能相应引脚的弱上拉或弱下拉功能。

当选择内部上拉电路时，引脚上拉为 VDD - 0.7V（典型值）。确保在使能了内部上拉电路时无外部上拉源，因为电压差会引起电流回路。

注：	只要端口引脚被配置为数字输出引脚，状态变化通知引脚上的弱上拉电路将始终被禁止。

例 9-1： 端口写 / 读示例

```
MOV    0xFF00, W0          ; Configure PORTB<15:8> as inputs
MOV    W0, TRISBB          ; and PORTB<7:0> as outputs
NOP                        ; Delay 1 cycle
BTSS   PORTB, #13          ; Next Instruction
```

 超前信息

9.4 外设引脚选择

通用器件的最大挑战是在最小化 I/O 引脚上功能冲突的同时，提供最大可能的外设功能集。在需要使用复用一个引脚的多个外设的应用中，对应用程序代码进行繁琐得更改或彻底重新设计可能是唯一的选择。

外设引脚选择功能通过使能用户外设集选择并将外设功能放置到大量的 I/O 引脚中提供了一个替代这两种选择的方案。通过增加特定器件上的引脚配置选项，用户可以更好的调节单片机以满足整个应用，而不是调整应用来满足器件。

外设引脚选择功能通过固定数量的数字 I/O 引脚进行操作。用户可将任一数字外设的输入和 / 或输出独立映射到这些 I/O 引脚之一。外设引脚选择通过软件执行，通常不需要对器件重新编程。此选择功能包括硬件保护机制，可用于在建立了外设映射后防止意外或误更改此映射。

9.4.1 可用引脚

外设引脚选择功能可在最多44个引脚的范围内使用，可用的引脚数量取决于特定器件及其引脚数。支持外设引脚选择功能的引脚在其完整引脚名称中有 “ RPn ” 或 “ RPIn ” 标识，“ n ” 指的是可重新映射的引脚的编号。“ RP ” 用于指定支持可重新映射输入及输出功能的引脚，“ RPI ” 则指定只支持可重新映射输入功能的引脚。

PIC24FJ256GA110 系列器件支持的可重新映射的只可用作输入的引脚数远大于其所支持的可重新映射输入／输出引脚数。在此系列器件中，根据选择的特定器件的引脚配置，最多有 32 个可重新映射输入 / 输出引脚，即从 RP0 至 P31。只支持可重新映射输入功能的引脚数范围为从 RPI32 到 RPI43 （或指定器件的上限）。

关于各封装提供的引脚布局，请参见表 1-4。

9.4.2 可用外设

可由外设引脚选择管理的外设都是数字外设。包括普通串行通信 （UART 和 SPI）、通用定时器时钟输入、定时器相关外设（输入捕捉和输出比较）以及外部中断输入。由于比较器模块的输出为离散数字信号，因此也包括在内。

外设引脚选择模块不适用于 I^2C™ 状态变化通知输入、RTCC 闹钟输出或具有模拟输入的外设。

有引脚选择和无引脚选择的外设的最大区别是有引脚选择的外设与默认 I/O 引脚之间无关联。必须在使用外设前将其分配给指定的 I/O 引脚。相反，假设无引脚选择的外设处于激活状态且未与其他外设发生冲突，则此外设始终在默认引脚上可用。

9.4.2.1 外设引脚选择功能优先级

可选引脚的外设在给定 I/O 引脚上激活时，它的优先级高于所有其他数字 I/O 和与该引脚相关的数字通信外设。给定的优先级与映射到的外设的类型无关。有引脚选择的外设的优先级始终低于与该引脚相关的任何模拟功能。

9.4.3 控制外设引脚选择

外设引脚选择功能是通过以下两组特殊功能寄存器控制的：一组用于映射外设输入，另一组用于映射输出。由于输入和输出是单独控制的，因此特定外设的输入和输出 （若该外设都有）均可施加到任何可选的功能引脚上，而没有限制。

根据映射的是输入还是输出，有两种方法可处理外设与外设可选引脚之间的关联。

9.4.3.1 输入映射

外设引脚选择选项的输入根据外设进行映射；即与外设相关的控制寄存器指示要映射到的引脚。RPINRx 寄存器用于配置外设输入映射（见寄存器 9-1 至寄存器 9-21）。各寄存器均包含两组 （各 6 位位域），其中每组都与一个可选引脚的外设相关。用适当的 6 位值编程给定外设的位域会将具有此值的 RPn 引脚映射到该外设。对于任何给定器件，任何位域值的有效范围对应于此器件所支持的外设引脚选择的最大值。

表 9-1： 可选输入源 （将输入映射到功能）[1]

输入名称	功能名称	寄存器	功能映射位
外部中断 1	INT1	RPINR0	INT1R5:INT1R0
外部中断 2	INT2	RPINR1	INT2R5:INT2R0
外部中断 3	INT3	RPINR1	INT3R5:INT3R0
外部中断 4	INT4	RPINR2	INT4R5:INT4R0
输入捕捉 1	IC1	RPINR7	IC1R5:IC1R0
输入捕捉 2	IC2	RPINR7	IC2R5:IC2R0
输入捕捉 3	IC3	RPINR8	IC3R5:IC3R0
输入捕捉 4	IC4	RPINR8	IC4R5:IC4R0
输入捕捉 5	IC5	RPINR9	IC5R5:IC5R0
输入捕捉 6	IC6	RPINR9	IC6R5:IC6R0
输入捕捉 7	IC7	RPINR10	IC7R5:IC7R0
输入捕捉 8	IC8	RPINR10	IC8R5:IC8R0
输入捕捉 9	IC9	RPINR15	IC9R5:IC9R0
输出比较故障 A	OCFA	RPINR11	OCFAR5:OCFAR0
输出比较故障 B	OCFB	RPINR11	OCFBR5:OCFBR0
SPI1 时钟输入	SCK1IN	RPINR20	SCK1R5:SCK1R0
SPI1 数据输入	SDI1	RPINR20	SDI1R5:SDI1R0
SPI1 从选择输入	SS1IN	RPINR21	SS1R5:SS1R0
SPI2 时钟输入	SCK2IN	RPINR22	SCK2R5:SCK2R0
SPI2 数据输入	SDI2	RPINR22	SDI2R5:SDI2R0
SPI2 从选择输入	SS2IN	RPINR23	SS2R5:SS2R0
SPI3 时钟输入	SCK3IN	RPINR23	SCK3R5:SCK3R0
SPI3 数据输入	SDI3	RPINR28	SDI3R5:SDI3R0
SPI3 从选择输入	SS3IN	RPINR29	SS3R5:SS3R0
Timer1 外部时钟	T1CK	RPINR2	T1CKR5:T1CKR0
Timer2 外部时钟	T2CK	RPINR3	T2CKR5:T2CKR0
Timer3 外部时钟	T3CK	RPINR3	T3CKR5:T3CKR0
Timer4 外部时钟	T4CK	RPINR4	T4CKR5:T4CKR0
Timer5 外部时钟	T5CK	RPINR4	T5CKR5:T5CKR0
UART1 允许发送	$\overline{\text{U1CTS}}$	RPINR18	U1CTSR5:U1CTSR0
UART1 接收	U1RX	RPINR18	U1RXR5:U1RXR0
UART2 允许发送	$\overline{\text{U2CTS}}$	RPINR19	U2CTSR5:U2CTSR0
UART2 接收	U2RX	RPINR19	U2RXR5:U2RXR0
UART3 允许发送	$\overline{\text{U3CTS}}$	RPINR21	U3CTSR5:U3CTSR0
UART3 接收	U3RX	RPINR17	U3RXR5:U3RXR0
UART4 允许发送	$\overline{\text{U4CTS}}$	RPINR27	U4CTSR5:U4CTSR0
UART4 接收	U4RX	RPINR27	U4RXR5:U4RXR0

注 1： 除非另外声明，否则所有输入均使用施密特触发器输入缓冲器。

9.4.3.2 输出映射

与输入相反，外设引脚选择选项的输出根据引脚进行映射。这种情况下，与特定引脚相关的控制寄存器指示要映射的外设输出。 RPORx 寄存器用于控制输出映射。各寄存器均包含两组 6 位位域，每组都与一个 RPn 引脚相关（见寄存器 9-22 至寄存器 9-37）。位域值对应一个外设，该外设的输出被映射到 RPn 引脚 （见表 9-2）。

由于采用的映射技术，输出映射的外设列表中还包含一个空值 “ 000000 ”。此值允许任何给定引脚始终与所有可选引脚的外设的输出断开。

表 9-2： 可选输出源 （将功能映射到输出）

输出功能编号 [1]	功能	输出名称
0	NULL[2]	空
1	C1OUT	比较器 1 输出
2	C2OUT	比较器 2 输出
3	U1TX	UART1 发送
4	$\overline{\text{U1RTS}}$[3]	UART1 请求发送
5	U2TX	UART2 发送
6	$\overline{\text{U2RTS}}$[3]	UART2 请求发送
7	SDO1	SPI1 数据输出
8	SCK1OUT	SPI1 时钟输出
9	SS1OUT	SPI1 从选择输出
10	SDO2	SPI2 数据输出
11	SCK2OUT	SPI2 时钟输出
12	SS2OUT	SPI2 从选择输出
18	OC1	输出比较 1
19	OC2	输出比较 2
20	OC3	输出比较 3
21	OC4	输出比较 4
22	OC5	输出比较 5
23	OC6	输出比较 6
24	OC7	输出比较 7
25	OC8	输出比较 8
28	U3TX	UART3 发送
29	$\overline{\text{U3RTS}}$[3]	UART3 请求发送
30	U4TX	UART4 发送
31	$\overline{\text{U4RTS}}$[3]	UART4 请求发送
32	SDO3	SPI3 数据输出
33	SCK3OUT	SPI3 时钟输出
34	SS3OUT	SPI3 从选择输出
35	OC9	输出比较 9
37-63	（未用）	NC

注 1： 使用列出的值设置 RPORx 寄存器，从而将输出功能分配给为相关的 RPn 引脚。
2： 在器件复位时将 NULL 功能分配给所有的 RPn 输出，并禁止 RPn 输出功能。
3： IrDA® BCLK 功能使用此输出。

9.4.3.3 映射限制

外设引脚选择的控制模式相当灵活。除阻止由两个配置为相同输入功能的物理引脚或配置为同一引脚的两个功能输出引起的信号冲突的系统电路外，无其他硬件强制的锁定。此灵活性可扩展到允许一个输入驱动多个外设或单功能输出驱动多个输出引脚。

9.4.3.4 PIC24FJ256GA110 系列器件的映射例外

虽然理论上PPS 寄存器最多允许 64 个可重新映射的 I/O 引脚，但是并不是全部器件都可以实现这些引脚。对于 PIC24FJ256GA110 系列器件而言，可用的可重新映射引脚的数量最多为 46 个，且其中包括 14 个只支持输入功能的引脚。另外， RP 和 RPI 序列中的一些引脚在引脚数较低的器件中未实现。表 9-3 中概述了可用的可重新映射引脚数的区别。

开发使用可重新映射引脚的应用时，用户应该注意以下几点：

- 对于 RPINRx 寄存器，与某一器件的未实现引脚对应的位组合将被视为无效处理；而相应的模块则不会将输入映射到该寄存器上。对于全部 PIC24FJ256GA110 系列器件而言，这包括所有大于 45 的值 （101101）。
- 对于 RPORx 寄存器，与未实现引脚相对应的位域也将不会被实现。写入这些位域将不会产生任何影响。

9.4.4 控制配置更改

由于在运行时外设重新映射可以被更改，因此需要对外设重新映射加以某些限制来阻止意外更改配置。PIC24F 器件有以下三种用于阻止更改外设映射的功能：

- 控制寄存器锁定序列
- 连续状态监视
- 配置位重新映射锁定

9.4.4.1 控制寄存器锁定

正常工作状态下，不允许写入 RPINRx 和 RPORx 寄存器。尝试的写入操作看似正常执行，但寄存器的内容并没有变化。要更改这些寄存器的内容，寄存器必须用硬件解锁。寄存器锁定由 IOLOCK 位 （OSCCON<6>）控制。置 1 IOLOCK 将阻止写入控制寄存器；而清零 IOLOCK 将允许写入。

要置 1 或清零 IOLOCK，必须执行以下指定命令序列：

1. 将 46h 写入 OSCCON<7:0>。
2. 将 57h 写入 OSCCON<7:0>。
3. 用一次操作清零 （或置 1） IOLOCK。

与振荡器 LOCK 位的类似序列不同，IOLOCK 在更改前一直保持一种状态。这允许所有外设引脚选择均可被配置为执行一个解锁序列，然后对所有控制寄存器进行更新，最后用第二个锁定序列锁定。

9.4.4.2 连续状态监视

除了阻止直接写入的保护外， RPINRx 和 RPORx 寄存器的内容还由影子寄存器通过硬件不停地进行监视。如果任何寄存器发生了不希望的更改 （例如，由 ESD 或其他外部事件引起的干扰），则将触发配置失配复位。

9.4.4.3 配置位引脚选择锁定

作为又一层保护，可配置器件以阻止对RPINRx和RPORx寄存器执行多次写会话。 IOL1WAY （CW2<4>）配置位阻止在 IOLOCK 位置 1 后再将其清零。若 IOLOCK 保持置 1 状态，寄存器解锁过程将不会执行，且不能写入外设引脚选择控制寄存器。清零该位并重新使能外设重新映射的唯一方法是执行器件复位。

默认 （未编程）状态下， IOL1WAY 置 1，限制用户只能进行一次写会话。编程 IOL1WAY 允许用户对外设引脚选择寄存器进行不受限制的访问（通过正确的使用解锁序列）。

表 9-3： PIC24FJ256GA110 系列器件的可重新映射引脚例外

器件引脚数	RP 引脚 （I/O）		RPI 引脚	
	总数	未实现	总数	未实现
64 引脚	29	RP5、 RP15 和 RP31	2	RPI32-36 和 RPI38-44
80 引脚	31	RP31	11	RPI32、 RPI39 和 RPI41
100 引脚	32	—	14	—

9.4.5 外设引脚选择注意事项

控制外设引脚选择的功能需要注意应用设计中几个容易被忽视的事项。这对于仅可用作可重新映射外设的几个常用外设尤为正确。

主要问题是在器件默认（复位）状态下外设引脚选择功能在默认引脚上不可用。由于所有 RPINRx 寄存器复位为 111111 且所有 RPORx 寄存器复位为 000000，因此所有外设引脚选择输入与 V_{SS} 相连，而所有外设引脚选择输出断开连接。

> **注：** 在将外设引脚选择输入连接到 RP63 时，器件上不必具有 RP63，因为寄存器复位就会实现这一设置。

此情景需要用户在执行任何其他应用程序代码前使用正确的外设配置初始化器件。由于 IOLOCK 位复位为解锁状态，所以不必在器件退出复位后执行解锁序列。但是，考虑到应用的安全性，最好在写入控制寄存器之后置 1 IOLOCK 位并锁定配置。

由于解锁序列对时序有严格要求，因此必须作为汇编语言程序执行，其工作方式与改变振荡器配置相同。若应用程序是用 C 或其他高级语言编写的，则解锁序列仍应通过编写行内汇编代码执行。

选择配置需要查看所有外设引脚选择及其引脚分配，尤其是那些未在应用中使用的外设。在所有情况下，未用的可选引脚的外设应完全被禁止。未用外设的输入应分配给未用的 RPn 引脚功能。具有未用的 RPn 功能的 I/O 引脚应配置为空外设输出。

将外设分配给特定引脚的操作不能自动执行引脚 I/O 电路的任何其他配置操作。理论上，也就是说将可选引脚的输出添加到某个引脚意味着在驱动输出时可能会无意识的驱动现有的外设输入。用户必须熟悉其他共用一个可重新映射引脚的固定外设的行为，了解何时使能或禁止这些外设。为了安全起见，共用一个引脚的固定数字外设在不用时应被禁止。

遵照这些方针，对特定外设的可重新映射的引脚进行配置并不会自动使能对应的外设。必须特别配置外设实现特定的操作并使能外设，如同将外设连接到固定引脚时一样。此操作在应用程序代码中的位置（紧跟在器件复位和外设配置之后或在主应用程序内）取决于外设及其在应用中的使用。

最后一个问题是外设引脚选择功能既不改写模拟输入也不会将具有模拟功能的引脚重新配置为数字 I/O。若某个引脚在器件复位时配置为模拟输入，则在使用该引脚的外设引脚选择功能时必须将此引脚明确重新配置为数字 I/O。

例 9-2 所示为使用 UART1 实现带有流控制的双向通信的配置。使用了以下输入和输出功能：

- 输入功能：U1RX 和 $\overline{\text{U1CTS}}$
- 输出功能：U1TX 和 $\overline{\text{U1RTS}}$

例 9-2： 配置 UART1 输入和输出功能

```
// Unlock Registers
asm volatile (  "MOV    #OSCCON, w1  \n"
                "MOV    #0x46, w2    \n"
                "MOV    #0x57, w3    \n"
                "MOV.b  w2, [w1]     \n"
                "MOV.b  w3, [w1]     \n"
                "BCLR OSCCON,#6");

// Configure Input Functions (Table 9-1))
    // Assign U1RX To Pin RP0
    RPINR18bits.U1RXR = 0;

    // Assign U1CTS To Pin RP1
    RPINR18bits.U1CTSR = 1;

// Configure Output Functions (Table 9-2)
    // Assign U1TX To Pin RP2
    RPOR1bits.RP2R = 3;

    // Assign U1RTS To Pin RP3
    RPOR1bits.RP3R = 4;

// Lock Registers
asm volatile (  "MOV    #OSCCON, w1  \n"
                "MOV    #0x46, w2    \n"
                "MOV    #0x57, w3    \n"
                "MOV.b  w2, <w1>     \n"
                "MOV.b  w3, <w1>     \n"
                "BSET   OSCCON, #6"  );
```

9.4.6 外设引脚选择寄存器

PIC24FJ256GA110 系列器件共有 37 个用于配置可重新映射的外设的寄存器：

- 输入可重新映射外设寄存器 （21）
- 输出可重新映射外设寄存器 （16）

注：	仅在 IOLOCK（OSCCON<6>）= 0 时才能改变输入和输出寄存器的值。请参见**第 9.4.4.1 节 “控制寄存器锁定”**了解特定的命令序列。

寄存器 9-1： RPINR0：外设引脚选择输入寄存器 0

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**INT1R5:INT1R0：**将外部中断 1 （INT1）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-0	**未实现：**读为 0

寄存器 9-2： RPINR1：外设引脚选择输入寄存器 1

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**INT3R5:INT3R0：**将外部中断 3 （INT3）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**INT2R5:INT2R0：**将外部中断 2 （INT2）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-3： RPINR2：外设引脚选择输入寄存器 2

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T1CKR5	T1CKR4	T1CKR3	T1CKR2	T1CKR1	T1CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**T1CKR5:T1CKR0：**将 Timer1 外部时钟 （T1CK）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**INT4R5:INT4R0：**将外部中断 4 （INT4）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-4： RPINR3：外设引脚选择输入寄存器 3

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**T3CKR5:T3CKR0：**将 Timer3 外部时钟 （T3CK）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**T2CKR5:T2CKR0：**将 Timer2 外部时钟 （T2CK）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-5： RPINR4：外设引脚选择输入寄存器 4

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T5CKR5	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T4CKR5	T4CKR4	T4CKR3	T4CKR2	T4CKR1	T4CKR0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**T5CKR5:T5CKR0：**将 Timer5 外部时钟 （T5CK）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**T4CKR5:T4CKR0：**将 Timer4 外部时钟 （T4CK）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-6： RPINR7：外设引脚选择输入寄存器 7

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC2R5	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC1R5	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**IC2R5:IC2R0：**将输入捕捉 2 （IC2）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**IC1R5:IC1R0：**将输入捕捉 1 （IC1）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-7： RPINR8：外设引脚选择输入寄存器 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC4R5	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC3R5	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 `0`

bit 13-8 **IC4R5:IC4R0：**将输入捕捉 4 （IC4）分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 **未实现：**读为 `0`

bit 5-0 **IC3R5:IC3R0：**将输入捕捉 3 （IC3）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-8： RPINR9：外设引脚选择输入寄存器 9

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC6R5	IC6R4	IC6R3	IC6R2	IC6R1	IC6R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC5R5	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 `0`

bit 13-8 **IC6R5:IC6R0：**将输入捕捉 6 （IC6）分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 **未实现：**读为 `0`

bit 5-0 **IC5R5:IC5R0：**将输入捕捉 5 （IC5）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-9： RPINR10：外设引脚选择输入寄存器 10

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC8R5	IC8R4	IC8R3	IC8R2	IC8R1	IC8R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC7R5	IC7R4	IC7R3	IC7R2	IC7R1	IC7R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**IC8R5:IC8R0：**将输入捕捉 8 （IC8）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**IC7R5:IC7R0：**将输入捕捉 7 （IC7）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-10： RPINR11：外设引脚选择输入寄存器 11

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**OCFBR5:OCFBR0：**将输出比较故障 B （OCFB）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**OCFAR5:OCFAR0：**将输出比较故障 A （OCFA）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-11： RPINR15：外设引脚选择输入寄存器 15

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC9R5	IC9R4	IC9R3	IC9R2	IC9R1	IC9R0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**IC9R5:IC9R0：**将输入捕捉 9 （IC9）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-0	**未实现：**读为 0

寄存器 9-12： RPINR17：外设引脚选择输入寄存器 17

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**U3RXR5:U3RXR0：**将 UART3 接收 （U3RX）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-0	**未实现：**读为 0

寄存器 9-13： RPINR18：外设引脚选择输入寄存器 18

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **U1CTSR5:U1CTSR0：**将 UART1 允许发送 （$\overline{\text{U1CTS}}$）分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 **未实现：**读为 0

bit 5-0 **U1RXR5:U1RXR0：**将 UART1 接收 （U1RX）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-14： RPINR19：外设引脚选择输入寄存器 19

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **U2CTSR5:U2CTSR0：**将 UART2 允许发送 （$\overline{\text{U2CTS}}$）分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 **未实现：**读为 0

bit 5-0 **U2RXR5:U2RXR0：**将 UART2 接收 （U2RX）分配给相应的 RPn 或 RPIn 引脚的位

 超前信息

寄存器 9-15： RPINR20：外设引脚选择输入寄存器 20

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**SCK1R5:SCK1R0：**将 SPI1 时钟输入 （SCK1IN）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**SDI1R5:SDI1R0：**将 SPI1 数据输入 （SDI1）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-16： RPINR21：外设引脚选择输入寄存器 21

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**U3CTSR5:U3CTSR0：**将 UART3 允许发送 （$\overline{\text{U3CTS}}$）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**SS1R5:SS1R0：**将 SPI1 从选择输入 （SS1IN）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-17： RPINR22：外设引脚选择输入寄存器 22

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**SCK2R5:SCK2R0：**将 SPI2 时钟输入 （SCK2IN）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**SDI2R5:SDI2R0：**将 SPI2 数据输入 （SDI2）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-18： RPINR23：外设引脚选择输入寄存器 23

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**SCK3R5:SCK3R0：**将 SPI3 时钟输入 （SCK3IN）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**SS2R5:SS2R0：**将 SPI2 从选择输入 （SS2IN）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-19： RPINR27：外设引脚选择输入寄存器 27

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U4CTSR5	U4CTSR4	U4CTSR3	U4CTSR2	U4CTSR1	U4CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U4RXR5	U4RXR4	U4RXR3	U4RXR2	U4RXR1	U4RXR0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**U4CTSR5:U4CTSR0：**将 UART4 允许发送 （$\overline{\text{U4CTS}}$）分配给相应的 RPn 或 RPIn 引脚的位
bit 7-6	**未实现：**读为 0
bit 5-0	**U4RXR5:U4RXR0：**将 UART4 接收 （U4RX）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-20： RPINR28：外设引脚选择输入寄存器 28

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-6	**未实现：**读为 0
bit 5-0	**SDI3R5:SDI3R0：**将 SPI3 数据输入 （SDI3）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-21： RPINR29：外设引脚选择输入寄存器 29

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-6 **未实现：**读为 0

bit 5-0 **SS3R5:SS3R0：**将 SPI3 从选择输入 （SS31IN）分配给相应的 RPn 或 RPIn 引脚的位

寄存器 9-22： RPOR0：外设引脚选择输出寄存器 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP1R5:RP1R0：**RP1 输出引脚映射位
将外设输出编号 n 分配给 RP1 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP0R5:RP0R0：**RP0 输出引脚映射位
将外设输出编号 n 分配给 RP0 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-23： RPOR1：外设引脚选择输出寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP3R5:RP3R0：**RP3 输出引脚映射位
将外设输出编号 n 分配给 RP3 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP2R5:RP2R0：**RP2 输出引脚映射位
将外设输出编号 n 分配给 RP2 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-24： RPOR2：外设引脚选择输出寄存器 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP5R5(1)	RP5R4(1)	RP5R3(1)	RP5R2(1)	RP5R1(1)	RP5R0(1)
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP5R5:RP5R0：**RP5 输出引脚映射位 (1)
将外设输出编号 n 分配给 RP5 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP4R5:RP4R0：**RP4 输出引脚映射位
将外设输出编号 n 分配给 RP4 引脚 （见表 9-2 了解外设功能编号）。

注 1： 在 64 引脚器件中未实现，读为 0。

寄存器 9-25： RPOR3：外设引脚选择输出寄存器 3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP7R5:RP7R0：**RP7 输出引脚映射位
将外设输出编号 n 分配给 RP7 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP6R5:RP6R0：**RP6 输出引脚映射位
将外设输出编号 n 分配给 RP6 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-26： RPOR4：外设引脚选择输出寄存器 4

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP9R5:RP9R0：**RP9 输出引脚映射位
将外设输出编号 n 分配给 RP9 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP8R5:RP8R0：**RP8 输出引脚映射位
将外设输出编号 n 分配给 RP8 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-27： RPOR5：外设引脚选择输出寄存器 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP11R5:RP11R0：**RP11 输出引脚映射位
将外设输出编号 n 分配给 RP11 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP10R5:RP10R0：**RP10 输出引脚映射位
将外设输出编号 n 分配给 RP10 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-28： RPOR6：外设引脚选择输出寄存器 6

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP13R5	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP12R5	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP13R5:RP13R0：**RP13 输出引脚映射位
将外设输出编号 n 分配给 RP13 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP12R5:RP12R0：**RP12 输出引脚映射位
将外设输出编号 n 分配给 RP12 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-29： RPOR7：外设引脚选择输出寄存器 7

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP15R5[(1)]	RP15R4[(1)]	RP15R3[(1)]	RP15R2[(1)]	RP15R1[(1)]	RP15R0[(1)]
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP15R5:RP15R0：**RP15 输出引脚映射位 [(1)]
将外设输出编号 n 分配给 RP15 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP14R5:RP14R0：**RP14 输出引脚映射位
将外设输出编号 n 分配给 RP14 引脚 （见表 9-2 了解外设功能编号）。

注 1： 在 64 引脚器件中未实现，读为 0。

 超前信息

寄存器 9-30： RPOR8：外设引脚选择输出寄存器 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP17R5:RP17R0：**RP17 输出引脚映射位
将外设输出编号 n 分配给 RP17 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP16R5:RP16R0：**RP16 输出引脚映射位
将外设输出编号 n 分配给 RP16 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-31： RPOR9：外设引脚选择输出寄存器 9

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP19R5:RP19R0：**RP19 输出引脚映射位
将外设输出编号 n 分配给 RP19 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP18R5:RP18R0：**RP18 输出引脚映射位
将外设输出编号 n 分配给 RP18 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-32： RPOR10：外设引脚选择输出寄存器 10

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**RP21R5:RP21R0：**RP21 输出引脚映射位 将外设输出编号 n 分配给 RP21 引脚 （见表 9-2 了解外设功能编号）。
bit 7-6	**未实现：**读为 0
bit 5-0	**RP20R5:RP20R0：**RP20 输出引脚映射位 将外设输出编号 n 分配给 RP20 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-33： RPOR11：外设引脚选择输出寄存器 11

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**RP23R5:RP23R0：**RP23 输出引脚映射位 将外设输出编号 n 分配给 RP23 引脚 （见表 9-2 了解外设功能编号）。
bit 7-6	**未实现：**读为 0
bit 5-0	**RP22R5:RP22R0：**RP22 输出引脚映射位 将外设输出编号 n 分配给 RP22 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-34： RPOR12：外设引脚选择输出寄存器 12

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP25R5:RP25R0：**RP25 输出引脚映射位
将外设输出编号 n 分配给 RP25 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP24R5:RP24R0：**RP24 输出引脚映射位
将外设输出编号 n 分配给 RP24 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-35： RPOR13：外设引脚选择输出寄存器 13

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13-8 **RP27R5:RP27R0：**RP27 输出引脚映射位
将外设输出编号 n 分配给 RP27 引脚 （见表 9-2 了解外设功能编号）。

bit 7-6 **未实现：**读为 0

bit 5-0 **RP26R5:RP26R0：**RP26 输出引脚映射位
将外设输出编号 n 分配给 RP26 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-36： RPOR14：外设引脚选择输出寄存器 14

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**RP29R5:RP29R0：**RP29 输出引脚映射位 将外设输出编号 n 分配给 RP29 引脚 （见表 9-2 了解外设功能编号）。
bit 7-6	**未实现：**读为 0
bit 5-0	**RP28R5:RP28R0：**RP28 输出引脚映射位 将外设输出编号 n 分配给 RP28 引脚 （见表 9-2 了解外设功能编号）。

寄存器 9-37： RPOR15：外设引脚选择输出寄存器 15

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP31R5(1)	RP31R4(1)	RP31R3(1)	RP31R2(1)	RP31R1(1)	RP31R0(1)
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP30R5(2)	RP30R4(2)	RP30R3(2)	RP30R2(2)	RP30R1(2)	RP30R0(2)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14	**未实现：**读为 0
bit 13-8	**RP31R5:RP31R0：**RP31 输出引脚映射位 (1) 将外设输出编号 n 分配给 RP31 引脚 （见表 9-2 了解外设功能编号）。
bit 7-6	**未实现：**读为 0
bit 5-0	**RP30R5:RP30R0：**RP30 输出引脚映射位 (2) 将外设输出编号 n 分配给 RP30 引脚 （见表 9-2 了解外设功能编号）。

注 1： 在 64 引脚和 80 引脚器件中未实现，读为 0。
2： 在 64 引脚器件中未实现，读为 0。

10.0 TIMER1

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的 " **第 14 章 定时器** "（DS39704A_CN）。

Timer1 模块是一个 16 位的定时器，可作为实时时钟（Real-Time Clock，RTC）的时间计数器，或作为自由运行的时段定时器 / 计数器。Timer1 可在以下三种模式下工作：

- 16 位定时器
- 16 位同步计数器
- 16 位异步计数器

Timer1 还支持以下特性：

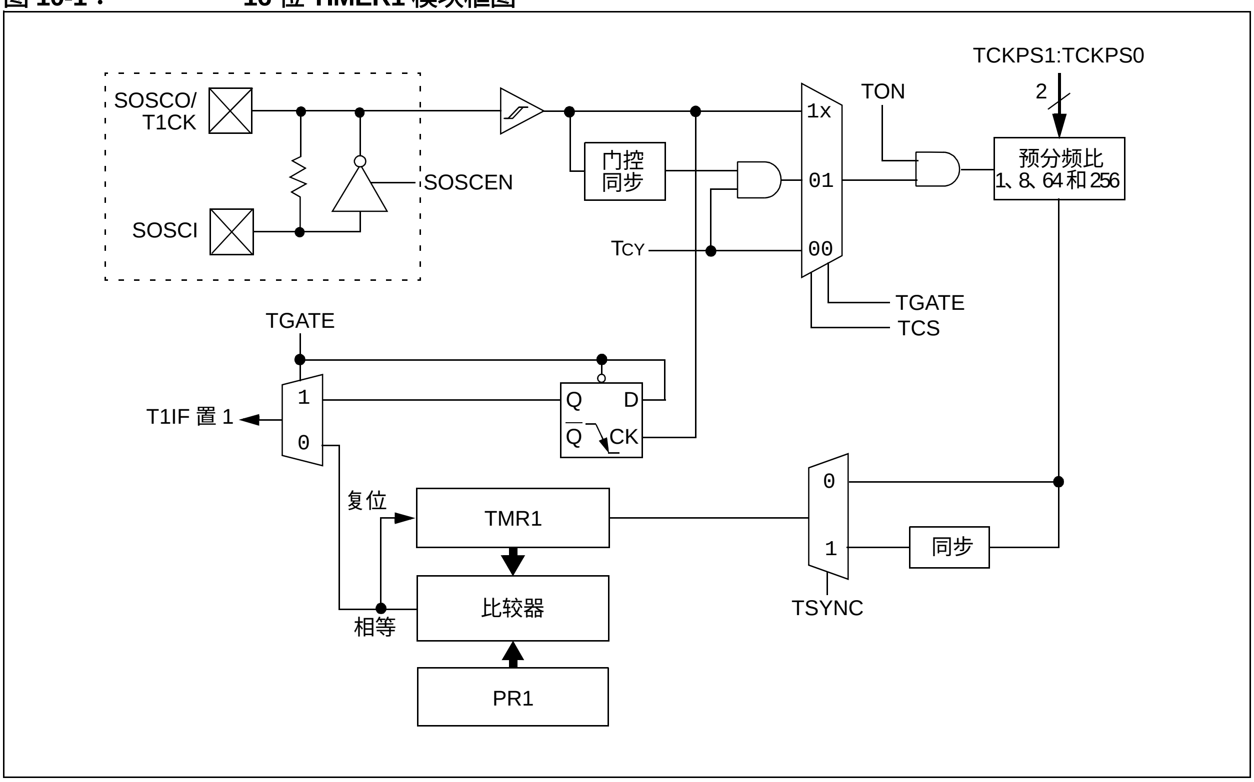
- 定时器门控操作
- 可选的预分频比设置
- CPU 空闲和休眠模式下的定时器操作
- 在 16 位周期寄存器匹配或外部门控信号的下降沿产生中断

图 10-1 给出了 16 位定时器模块的框图。

配置 Timer1 的操作：

1. 将 TON 位置 1 （= 1）。
2. 使用TCKPS1:TCKPS0位选择定时器预分频比。
3. 使用 TCS 和 TGATE 位设置时钟和门控模式。
4. 将TSYNC位置1或清零以配置同步或异步操作。
5. 将定时器周期值装入 PR1 寄存器。
6. 如果需要中断，则将中断允许位 T1IE 置 1。使用优先级位 T1IP2:T1IP0 来设置中断优先级。

图 10-1： 16 位 TIMER1 模块框图



寄存器 10-1： T1CON：TIMER1 控制寄存器 (1)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
bit7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **TON：**Timer1 启动位
1 = 启动 16 位 Timer1
0 = 停止 16 位 Timer1

bit 14 **未实现：**读为 0

bit 13 **TSIDL：**空闲模式下停止位
1 = 器件进入空闲模式后模块停止工作
0 = 在空闲模式下模块继续工作

bit 12-7 **未实现：**读为 0

bit 6 **TGATE：**Timer1 门控时间累加使能位
当 TCS = 1 时：
忽略此位。
当 TCS = 0 时：
1 = 使能门控时间累加
0 = 禁止门控时间累加

bit 5-4 **TCKPS1:TCKPS0：**Timer1 输入时钟预分频比选择位
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **未实现：**读为 0

bit 2 **TSYNC：**Timer1 外部时钟输入同步选择位
当 TCS = 1 时：
1 = 同步外部时钟输入
0 = 不同步外部时钟输入
当 TCS = 0 时：
忽略此位。

bit 1 **TCS：**Timer1 时钟源选择位
1 = T1CK 引脚的外部时钟的上升沿
0 = 内部时钟 （Fosc/2）

bit 0 **未实现：**读为 0

注 1： 在定时器运行 （TON = 1）时更改 TMRxCON 的值会导致定时器预分频计数器复位，所以不推荐这样做。

11.0 TIMER2/3 和 TIMER4/5

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见 《PIC24F 系列参考手册》中的 “**第 14 章 定时器** ”（DS39704A_CN）。

Timer2/3 和 Timer4/5 模块都是 32 位定时器，它们也可以被配置为四个独立的、工作模式可选的 16 位定时器。

作为 32 位定时器，Timer2/3 和 Timer4/5 可以工作在以下三种模式：

- 两个独立的 16 位定时器具有全部 16 位工作模式（异步计数器模式除外）
- 一个 32 位定时器
- 一个 32 位同步计数器

它们还支持以下功能：

- 定时器门控操作
- 可选的预分频比设置
- 在空闲模式和休眠模式期间的定时器工作
- 在 32 位周期寄存器匹配时产生中断
- ADC 事件触发信号 （仅 Timer4/5）

在作为 16 位定时器时，所有四个定时器都可以单独用作同步定时器或计数器。它们也可以提供上面列出的功能，但 ADC 事件触发除外，它只能在 Timer5 上实现。工作模式和是否使能由在 T2CON、 T3CON、 T4CON 和 T5CON 寄存器中相应位的设置决定。寄存器 11-1 给出了 T2CON 和 T4CON 的一般形式；寄存器 11-2 给出了 T3CON 和 T5CON 的一般形式。

当工作在 32 位定时器 / 计数器模式时，Timer2 和 Timer4 是最低有效字；而 Timer3 和 Timer5 是 32 位定时器的最高有效字。

注：	当定时器工作在32位模式时，忽略T3CON 和 T5CON 中的控制位。只有 T2CON 和 T4CON 控制位用于设置和控制。Timer2 和 Timer4 时钟和门控输入用于 32 位定时器模块，但产生中断时会将 Timer3 或 Timer5 的中断标志置 1。

将 Timer2/3 或 Timer4/5 配置为 32 位工作模式：

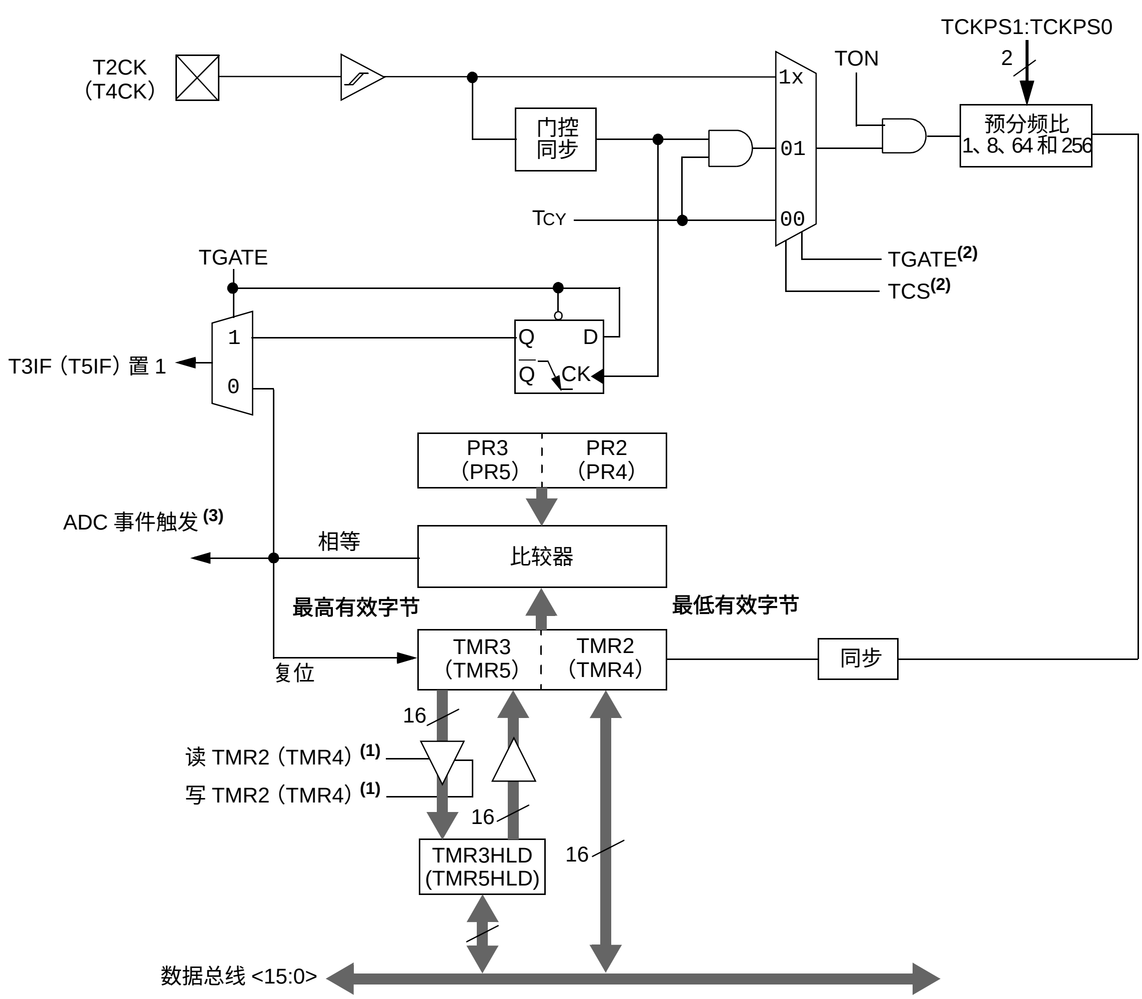
1. 将 T32 位（T2CON<3> 或 T4CON<3> = 1）置 1。
2. 使用 TCKPS1:TCKPS0 位选择 Timer2 或 Timer4 的预分频比。
3. 使用 TCS 和 TGATE 位设置时钟和门控模式。如果 TCS 被设置为外部时钟，则必须将 RPINRx（TxCK）配置给一个可用的 RPn 引脚。具体信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。
4. 装载定时器周期值。PR3（或 PR5）将包含值的最高有效字，而 PR2 （或 PR4）将包含值的最低有效字。
5. 如果需要中断，将中断允许位 T3IE 或 T5IE 置 1，并使用优先级位 T3IP2:T3IP0 或 T5IP2:T5IP0 设置中断优先级。请注意虽然定时器是由 Timer2 或 Timer4 控制，但其中断却表现为 Timer3 或 Timer5 中断。
6. 将 TON 位置 1 （= 1）。

定时器的值可以随时存储到 TMR3:TMR2（或 TMR5:TMR4）寄存器对中。TMR3 （TMR5）始终存储计数值的最高有效字，而 TMR2 （TMR4）始终放最低有效字。

将任何定时器配置为独立的 16 位工作模式：

1. 将和该定时器相应的 T32 位清零 （对于 Timer2 和 Timer3 来说是 T2CON<3> ；对于 Timer4 和 Timer5 来说是 T4CON<3>）。
2. 使用 TCKPS1:TCKPS0 位选择定时器预分频比。
3. 使用 TCS 和 TGATE 位设置时钟和门控模式。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。
4. 将定时器周期值装入 PRx 寄存器。
5. 如果需要中断，将中断允许位 TxIE 置 1，并使用优先级位 TxIP2:TxIP0 设置中断优先级。
6. 将 TON 位置 1 （TxCON<15> = 1）。

图 11-1： TIMER2/3 和 TIMER4/5 （32 位）框图



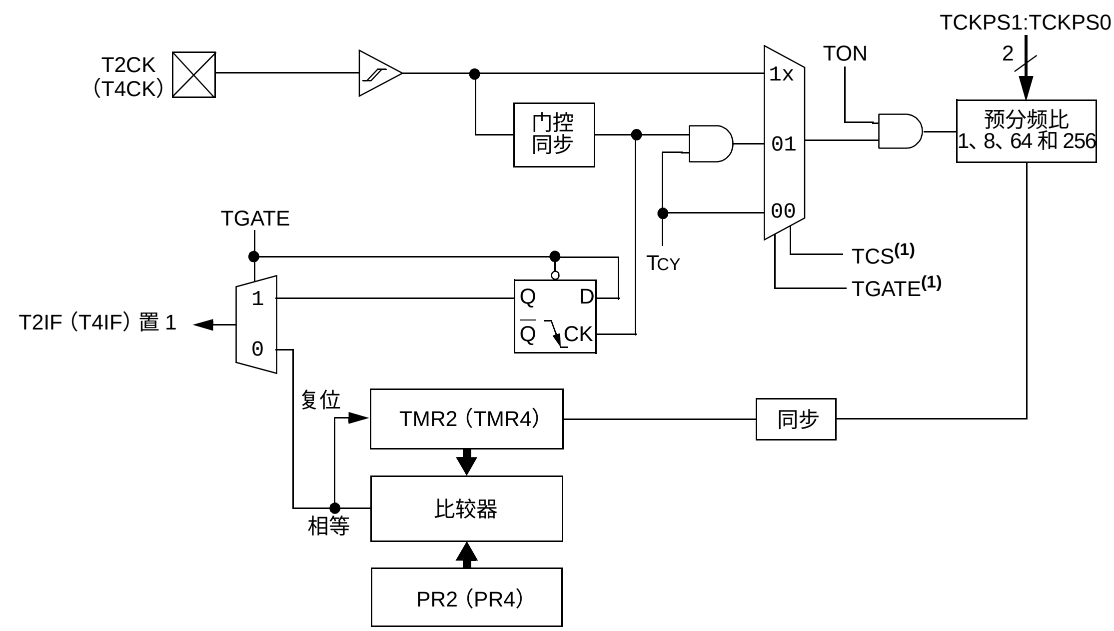
注 1： 必须将 32 位定时器配置位 T32 置 1 才能使定时器工作在 32 位定时器 / 计数器模式下。所有控制位分别在 T2CON 寄存器和 T4CON 寄存器中。

2： 在使用前，定时器时钟输入必须分配给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

3： 只有 Timer4/5 可以产生 ADC 事件触发信号。

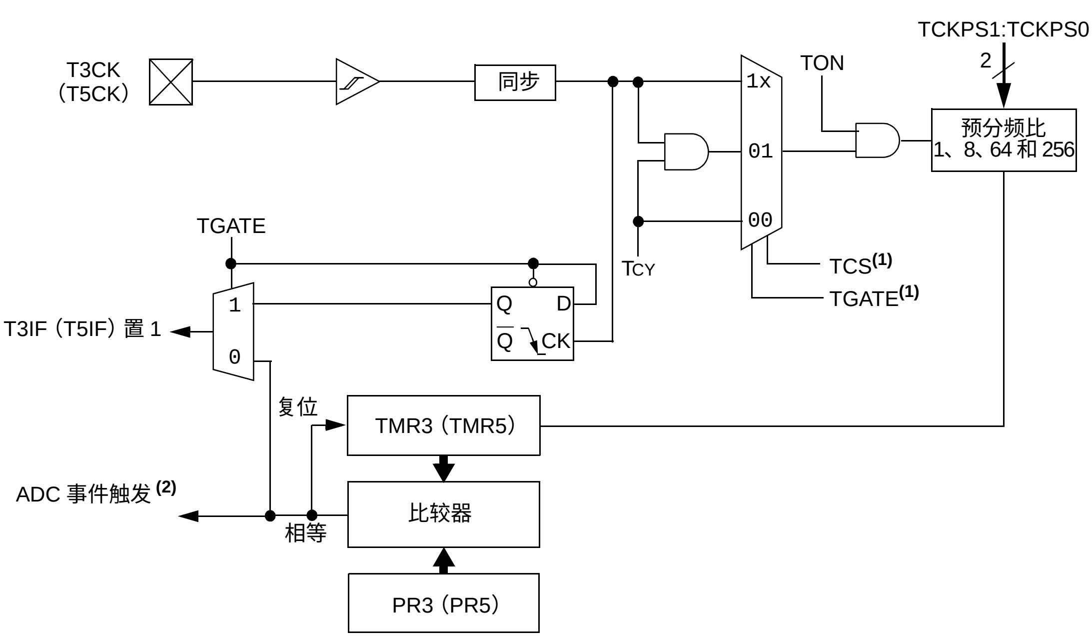
 超前信息

图 11-2：　　TIMER2 和 TIMER4 （16 位同步）框图



注　1： 在使用前，定时器时钟输入必须分配给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

图 11-3：　　TIMER3 和 TIMER5 （16 位异步）框图



注　1： 在使用前，定时器时钟输入必须分配给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。
　　2： 只有 Timer5 可以产生 ADC 事件触发信号。

寄存器 11-1： TxCON：TIMER2 和 TIMER4 控制寄存器[3]

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	T32[1]	—	TCS[2]	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **TON：**Timerx 启动位
<u>当 TxCON<3> = 1 时：</u>
1 = 启动 32 位 Timerx/y
0 = 停止 32 位 Timerx/y
<u>当 TxCON<3> = 0 时：</u>
1 = 启动 16 位 Timerx
0 = 停止 16 位 Timerx

bit 14 **未实现：**读为 0

bit 13 **TSIDL：**空闲模式下停止位
1 = 器件进入空闲模式模块停止工作
0 = 空闲模式下模块继续工作

bit 12-7 **未实现：**读为 0

bit 6 **TGATE：**Timerx 门控时间累加使能位
<u>当 TCS = 1 时：</u>
忽略此位。
<u>当 TCS = 0 时：</u>
1 = 使能门控时间累加
0 = 禁止门控时间累加

bit 5-4 **TCKPS1:TCKPS0：**Timerx 输入时钟预分频比选择位
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **T32：**32 位定时器模式选择位[1]
1 = Timerx 和 Timery 构成一个 32 位定时器
0 = Timerx 和 Timery 作为两个 16 位定时器
在 32 位模式下， T3CON 控制位不影响 32 位定时器的工作。

bit 2 **未实现：**读为 0

bit 1 **TCS：**Timerx 时钟源选择位[2]
1 = 来自 TxCK 引脚的外部时钟 （上升沿触发）
0 = 内部时钟 （Fosc/2）

bit 0 **未实现：**读为 0

注 1： 在 32 位模式下， T3CON 或 T5CON 控制位不影响 32 位定时器的工作。
2： 如果 TCS = 1，必须将 RPINRx （TxCK）配置给一个可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。
3： 在定时器运行时 （TON = 1）更改 TMRxCON 的值会导致定时器预分频计数器复位，所以不推荐这样做。

寄存器 11-2： TyCON：TIMER3 和 TIMER5 控制寄存器 (3)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON(1)	—	TSIDL(1)	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE(1)	TCKPS1(1)	TCKPS0(1)	—	—	TCS(1,2)	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **TON**：Timery 启动位 (1)
1 = 启动 16 位 Timery
0 = 停止 16 位 Timery

bit 14 **未实现**：读为 0

bit 13 **TSIDL**：空闲模式下停止位 (1)
1 = 器件进入空闲模式后模块停止工作
0 = 空闲模式下模块继续工作

bit 12-7 **未实现**：读为 0

bit 6 **TGATE**：Timery 门控时间累加使能位 (1)
当 TCS = 1 时：
忽略此位。
当 TCS = 0 时：
1 = 使能门控时间累加
0 = 禁止门控时间累加

bit 5-4 **TCKPS1:TCKPS0**：Timery 输入时钟预分频比选择位 (1)
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3-2 **未实现**：读为 0

bit 1 **TCS**：Timery 时钟源选择位 (1,2)
1 = 来自 TyCK 引脚的外部时钟 （上升沿触发）
0 = 内部时钟 （FOSC/2）

bit 0 **未实现**：读为 0

注 1： 当使能 32 位工作模式时 （T2CON<3> 或 T4CON<3> = 1），这些位不会对 Timery 的工作产生影响。定时器的所有功能都通过 T2CON 和 T4CON 寄存器设置。

2： 如果 TCS = 1，必须将 RPINRx （TxCK）配置给一个可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

3： 在定时器运行时 （TON = 1）更改 TMRxCON 的值会导致定时器预分频计数器复位，所以不推荐这样做。

注：

12.0 使用专用定时器的输入捕捉

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》。

PIC24FJ256GA110 系列中的所有器件都具有 9 个独立的增强型输入捕捉模块。每个模块都会提供大量配置及操作选项，用于捕捉外部脉冲事件和产生中断。

该增强型输出模块的主要特性包括：

- 通过级联相邻的两个模块，可通过硬件配置为所有的 32 位工作模式
- 输出比较操作有同步和触发两种模式，最多有 30 个用户可选的触发 / 同步源可供使用
- 用于捕捉和保持多个事件的定时器值的 4 级 FIFO 缓冲器
- 产生可配置中断
- 每个模块最多具有 6 个可用的时钟源，可驱动独立的内部 16 位计数器

该模块由 ICxCON1 （寄存器 12-1）和 ICxCON2 （寄存器 12-2）两个寄存器控制。图 12-1 所示为该模块的一般框图。

12.1 一般工作模式

12.1.1 同步和触发模式

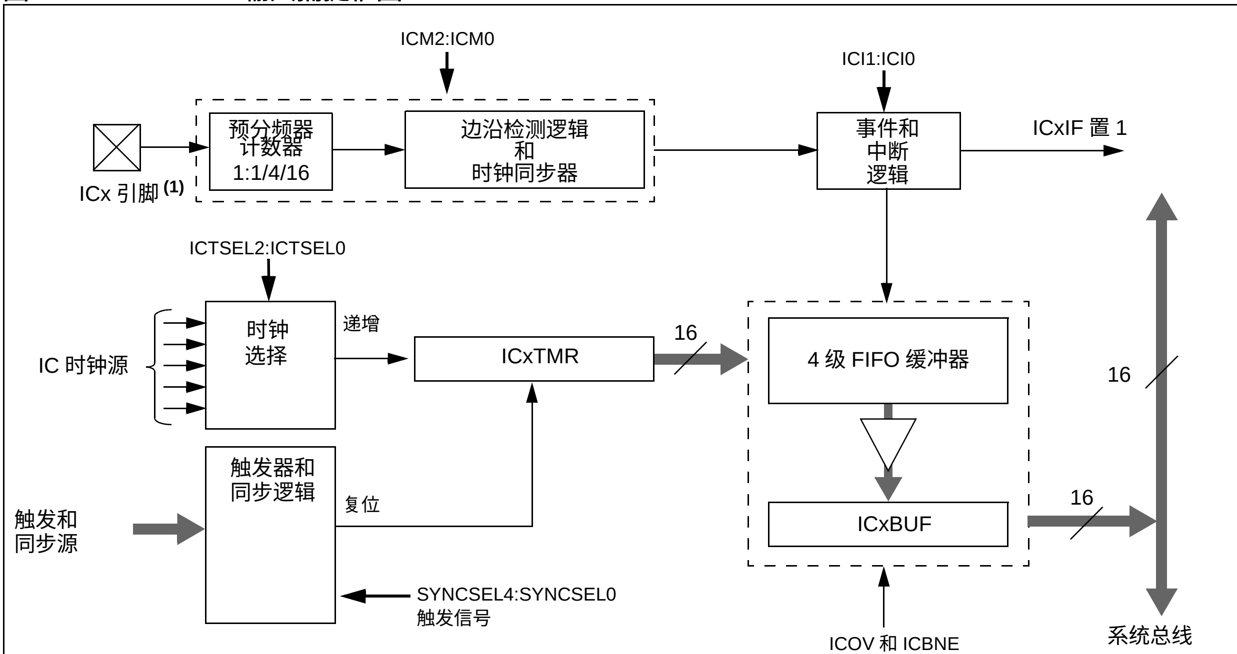
默认情况下，增强型输入捕捉模块工作在自由运行模式下。内部 16 位计数器 ICxTMR 连续递增计数，每次溢出时从 FFFFh 返回到 0000h，其周期与所选外部时钟源同步。发生捕捉事件时，内部计数器当前的 16 位值会被写入 FIFO 缓冲器。

在同步模式下，一旦使能选中的时钟源，模块就开始捕捉 ICx 引脚上的事件。选中的同步源上一发生事件，内部计数器就会复位。在触发模式下，模块会在允许内部计数器运行之前等待来自其余内部模块的同步事件发生。

通过将 SYNCSEL 位设置为 `00000`，并清零 ICTRIG 位（ICxCON2<7>），可选择工作在标准的自由运行模式下。SYNCSEL 位任何时候被设置为除 `00000` 以外的任何值，都可以选择同步和触发模式。 ICTRIG 位决定选择的是同步还是触发模式；将该位置 1 选择工作在触发模式下。在这两种模式中，由 SYNCSEL 位确定同步 / 触发源。

当 SYNCSEL 位设置为 `00000` 且 ICTRIG 位置 1 时，模块工作在软件触发模式下。这种情况下，通过手动置 1 TRIGSTAT 位 （ICxCON2<6>）启动捕捉操作。

图 12-1： 输入捕捉框图



注 1： 在使用前， ICx 输入必须分配给可用的 RPn 引脚。欲知更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

12.1.2 级联 （32 位）模式

默认情况下，每个模块都以其自身的 16 位定时器独立运行。要提高分辨率，可以将相邻的奇偶模块配置为一个 32 位模块。（例如，模块 1 和 2 配对，模块 3 和 4 配对等。）编号为奇数的模块 （ICx）提供 32 位寄存器对的低 16 位，偶数模块 （ICy）提供高 16 位。 ICx 寄存器每次溢出时从 FFFFh 返回到 0000h 的操作会使相应的 ICy 寄存器加 1。

通过将两个模块的 IC32 位（ICxCON2<8>）置 1 使用硬件配置级联操作。

对于 32 位级联操作，设置过程有些不同：

1. 将两个模块的 IC32 位（ICyCON2<8> 和 ICxCON2<8>）置 1，首先使能编号为偶数的模块。这可以确保这两个模块同时启动。
2. 将两个模块的 ICTSEL 和 SYNCSEL 位置 1 以选择相同的同步 / 触发源以及时基源。先设置偶数模块，再设置奇数模块。两个模块必须使用相同的 ICTSEL 和 SYNCSEL 设置。
3. 将偶数模块的 ICTRIG 位 （ICyCON2<7>）清零，这将强制该模块以与奇数模块同步的模式运行，不管其触发设置如何。
4. 使用奇数模块的 ICI 位 （ICxCON1<6:5>）设置所需的中断频率。
5. 使用奇数模块的 ICTRIG 位（ICxCON2<7>）配置工作在触发模式或同步模式。

注：	同步模式操作使能同步源作为最后一个步骤。两种输入捕捉模块都将保持复位状态直到使能同步源。

6. 使用奇数模块的 ICM 位（ICxCON1<2:0>）以设置所需的捕捉模式。

当使能了时基和触发 / 同步源时，该模块为捕捉事件做好了准备。当 ICBNE 位（ICxCON1<3>）置 1 时，FIFO 缓冲器中至少有一个捕捉值。读 FIFO 中的输入捕捉值直到 ICBNE 清零。

对于 32 位操作，读 ICxBUF 和 ICyBUF（ICxBUF 用于低 16 位， ICyBUF 用于高 16 位）获取完整的 32 位定时器值。当奇数模块的 ICBNE 位 （ICxCON1<3>）置 1 时，FIFO 缓冲器中至少有一个捕捉值。持续读缓冲器寄存器直到 ICBNE 位清零 （由硬件自动执行）。

12.2 捕捉操作

增强型输入捕捉模块可配置为在 ICx 的上升沿或所有跳变沿处捕捉定时器值并产生中断。捕捉也可配置为在所有上升沿或仅在某些 （每 4 个或 16 个）上升沿发生。可单独配置中断，使之在每个事件发生时，或者在一系列事件发生之后产生。

设置模块以进行捕捉操作：

1. 将 ICx 输入配置为可用外设引脚选择引脚中的一个。
2. 如果使用同步模式，在开始前禁止同步源。
3. 确保通过读 ICxBUF 删除了 FIFO 中的所有先前数据，直到 ICBNE 位 （ICxCON1<3>）清零。
4. 设置 SYNCSEL 位（ICxCON2<4:0>）来获取所需的同步 / 触发源。
5. 设置 ICTSEL 位 （ICxCON1<12:10>）来获取所需的时钟源。
6. 设置 ICI 位 （ICxCON1<6:5>）来获取所需的中断频率。
7. 选择同步或触发模式操作：
   a) 确认 SYNCSEL 位未设置为 `00000`。
   b) 对同步模式，清零 ICTRIG 位 （ICxCON2<7>）。
   c) 对触发模式，将 ICTRIG 置 1， TRIGSTAT 位 （ICxCON2<6>）清零。
8. 设置 ICM 位（ICxCON1<2:0>）来获取所需的工作模式。
9. 使能选中的触发 / 同步源。

 超前信息

寄存器 12-1：　ICxCON1：输入捕捉 x 控制寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R-0， HC	R-0， HC	R/W-0	R/W-0	R/W-0
—	ICI1	ICI0	ICOV	ICBNE	ICM2(1)	ICM1(1)	ICM0(1)
bit 7							bit 0

图注：	HC = 可由硬件清零的位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14　**未实现：**读为 0

bit 13　**ICSIDL：**空闲模式输入捕捉 x 模块停止控制位
1 = 输入捕捉模块将在 CPU 空闲模式下停止工作
0 = 在 CPU 空闲模式下输入捕捉模块继续工作

bit 12-10　**ICTSEL2:ICTSEL0：**输入捕捉定时器选择位
111 = 系统时钟 （Fosc/2）
110 = 保留
101 = 保留
100 = Timer1
011 = Timer5
010 = Timer4
001 = Timer2
000 = Timer3

bit 9-7　**未实现：**读为 0

bit 6-5　**ICI1:ICI0：**选择每次中断发生的捕捉次数的位
11 = 每四个捕捉事件中断一次
10 = 每三个捕捉事件中断一次
01 = 每两个捕捉事件中断一次
00 = 每个捕捉事件中断一次

bit 4　**ICOV：**输入捕捉 x 溢出状态标志位 （只读）
1 = 发生了输入捕捉溢出
0 = 未发生输入捕捉溢出

bit 3　**ICBNE：**输入捕捉 x 缓冲器空状态位 （只读）
1 = 输入捕捉缓冲器非空，至少还有一个以上的捕捉值
0 = 输入捕捉缓冲器为空

bit 2-0　**ICM2:ICM0：**输入捕捉模式选择位 (1)
111 = 中断模式：只有当器件处于休眠或空闲状态时输入捕捉才可作为中断引脚 （只检测上升沿，其余的控制位不适用）
110 = 未使用 （禁止模块）
101 = 预分频器捕捉模式：每 16 个上升沿捕捉一次
100 = 预分频器捕捉模式：每 4 个上升沿捕捉一次
011 = 简单捕捉模式：每个上升沿捕捉一次
010 = 简单捕捉模式：每个下降沿捕捉一次
001 = 边沿检测捕捉模式：每个沿 （上升沿或下降沿）捕捉一次， ICI1:ICI0 位不会控制该模式下的中断产生
000 = 输入捕捉模块关闭

注　1： ICx 输入必须配置给可用的 RPn 引脚。欲知更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

寄存器 12-2： ICxCON2：输入捕捉 x 控制寄存器 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	IC32
bit 15							bit 8

R/W-0	R/W-0, HS	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

图注：	HS = 可由硬件置 1 的位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-9 **未实现：**读为 0

bit 8 **IC32：**级联两个 IC 模块使能位 （32 位操作）
1 = ICx 和 ICy 级联成 32 位模块 （两个模块中的该位都必须置 1）
0 = ICx 单独用作 16 位模块

bit 7 **ICTRIG：**ICx 触发 / 同步选择位
1 = 由 SYNCSELx 位指定的源触发 ICx
0 = 由 SYNCSELx 位指定的源同步 ICx

bit 6 **TRIGSTAT：**定时器触发状态位
1 = 定时器源被触发并运行 （可以由硬件置 1，或使用软件置 1）
0 = 定时器源未被触发并保持清零状态

bit 5 **未实现：**读为 0

bit 4-0 **SYNCSEL4:SYNCSEL0：**触发 / 同步源选择位
11111 = 保留
11110 = 输入捕捉 9
11101 = 输入捕捉 6
11100 = CTMU(1)
11011 = A/D(1)
11010 = 比较器 3(1)
11001 = 比较器 2(1)
11000 = 比较器 1(1)
10111 = 输入捕捉 4
10110 = 输入捕捉 3
10101 = 输入捕捉 2
10100 = 输入捕捉 1
10011 = 输入捕捉 8
10010 = 输入捕捉 7
1000x = 保留
01111 = Timer 5
01110 = Timer 4
01101 = Timer 3
01100 = Timer 2
01011 = Timer 1
01010 = 输入捕捉 5
01001 = 输出捕捉 9
01000 = 输出比较 8
00111 = 输出比较 7
00110 = 输出比较 6
00101 = 输出比较 5
00100 = 输出比较 4
00011 = 输出比较 3
00010 = 输出比较 2
00001 = 输出比较 1
00000 = 不与任何模块同步

注 1： 这些输入只用作触发源，从不用作同步源。

13.0 带专用定时器的输出比较

> 注： 本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》。

PIC24FJ256GA110 系列器件都有 9 个独立的增强输出比较模块。其中每一个模块都提供了大量配置和工作选项，以在发生器件内部事件时产生连续的脉冲，并可为驱动功率应用产生脉宽调制波形。

增强输出比较模块的主要特性包括：

- 通过级联两个相邻的模块，可通过硬件配置为所有 32 位工作模式
- 输出比较操作有同步和触发两种模式，最多有 30 个用户可选的触发 / 同步源可供使用
- 两个单独的周期寄存器 （一个主寄存器 OCxR 和一个辅助寄存器 OCxRS）为产生脉宽可变的脉冲提供了更大的灵活性
- 配置在发生输出事件时产生单脉冲或连续脉冲，或者产生连续的 PWM 波形
- 每个模块最多有 6 种时钟源可供使用，可驱动一个单独的内部 16 位计数器

13.1 一般工作模式

13.1.1 同步和触发模式

默认情况下，增强输出比较模块工作于自由运行模式。内部 16 位计数器 OCxTMR 连续递增计数，每次溢出时从 FFFFh 折回 0000h，其周期与外部时钟源同步。内部计数器的值与周期寄存器的值每匹配一次，就会发生一次比较或 PWM 事件。

在同步模式下，只要选定的时钟源被使能，该模块就启动比较或 PWM 操作。只要选定的同步源发生事件，该模块的内部计数器就会复位。在触发模式下，该模块在允许计数器运行之前会等待来自另外一个内部模块的同步事件。

默认情况下或 SYNCSEL 位 （OCxCON2<4:0>）被设置为 00000 时，选择自由运行模式。SYNCSEL 位设置为除 00000 外的任何值时，选择同步或触发模式。可通过 OCTRIG 位 （OCxCON2<7>）选择同步模式或触发模式；将该位置 1 选择触发模式操作。在两种模式下，SYNCSEL 位都可决定同步 / 触发源。

13.1.2 级联 （32 位）模式

默认情况下，每种模式都以其自身的一组 16 位定时器和占空比寄存器独立运行。为了提高分辨率，可将相邻的偶数和奇数模块配置为一个 32 位的模块。（例如，模块 1 和模块 2 配对，模块 3 和模块 4 配对等。）奇数模块（OCx）为 32 位寄存器对提供低 16 位，而偶数模块（OCy）提供高 16 位。 OCx 寄存器的折回操作会引起其相应的 OCy 寄存器加 1。

对这两种模块而言，级联操作都通过将 OC32 位（OCxCON2<8>）置 1 在硬件内完成配置。

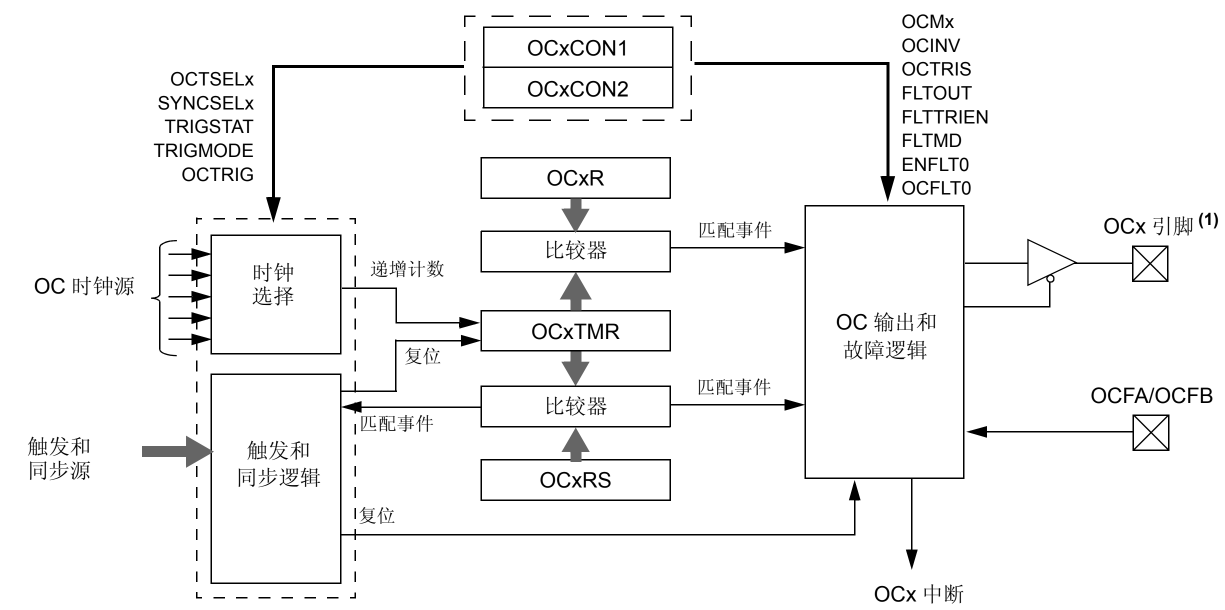
13.2 比较操作

在比较模式下 （图 13-1），可以配置增强输出比较模块，以产生单脉冲或连续的脉冲；也可以在每次定时器事件中，将输出引脚不断翻转。

为比较操作设置模块：

1. 将 OCx 输出配置为可用的外设引脚选择引脚之一。
2. 计算 OCxR 和 （对于双比较模式） OCxRS 占空比寄存器所需的值。
   a） 确定指令时钟周期的时间。考虑定时器源（如果使用了一个定时器源）的外部时钟频率和定时器预分频比的设置。
   b） 计算从定时器起始值 （0000h）到输出脉冲的上升沿所需的时间。
   c） 根据所需的脉冲宽度和到脉冲上升沿的时间，计算出现脉冲下降沿的时间。
3. 将上升沿的值写入 OCxR，将下降沿的值写入 OCxRS。
4. 将定时器周期寄存器PRy的值设置为大于或等于 OCxRS 中的值。
5. 将 OCM2:OCM0 位置 1，以选择适当的比较操作（= 0xx）。
6. 对于触发模式操作，将 OCTRIG 置 1 可使能触发模式。将 TRIGMODE 置 1 或清零可配置触发操作，将 TRIGSTAT 置 1 可选择硬件触发，将 TRIGSTAT 清零可选择软件触发。对于同步模式，可将 OCTRIG 清零。
7. 设置 SYNCSEL4:SYNCSEL0 位可配置触发或同步源。如果要求自由运行的定时器操作，可将 SYNCSEL位设置为00000（没有同步/触发源）。
8. 使用OCTSEL2:OCTSEL0位可选择时基源。如果有必要的话，将使能比较时基进行计数的选定定时器的 TON 位置 1。同步模式操作在时基使能时启动；而触发模式操作在发生了一个触发源事件后才启动。

图 13-1： 输出比较框图 （16 位模式）



注 1： 在使用之前， OCx 输出必须连接到可用的 RPn 引脚。欲知更多信息，请参见**第 9.4 节 “外设引脚选择”**。

对于 32 位级联操作，这些也是必要的步骤：

1. 将两个寄存器（OCyCON2<8> 和（OCxCON2<8>）的 OC32 位置 1。先使能偶数编号的模块，以确保奇偶模块同时启动。
2. 将偶数模块（OCyCON2）的 OCTRIG 位清零，这样模块将在同步模式下运行。
3. 对 OCyCON2 所需的输出和故障设置进行配置。
4. 通过将 OCTRIS 位置 1 强制 OCx 的输出引脚为输出状态。
5. 如果需要触发模式操作，通过设置 OCTRIG（OCxCON2<7>）、TRIGSTAT（OCxCON2<6>）和 SYNCSEL（OCxCON2<4:0>）位对 OCx 的触发选项进行配置。
6. 先为 OCyCON1 配置所要求的比较或 PWM 模式（通过设置 OCM<2:0>），再对 OCxCON1 进行此配置。

根据选择的输出模式，模块将使 OCx 引脚保持默认状态，并在 OCxR 和定时器的数据匹配时强行翻转引脚的状态。在双比较模式下，当 OCxRS 与定时器的数据匹配时 OCx 将重新回到默认状态。OCxIF 的中断标志位在 OCxR 匹配（单比较模式下）和每次 OCxRS 匹配（双比较模式下）时被置 1。

单脉冲事件只发生一次，但是可以通过简单地重写 OCxCON1 寄存器的值使单脉冲事件重复发生。连续脉冲事件会无限地持续进行直到被终止。

13.3 脉宽调制（PWM）模式

在 PWM 模式下，增强的输出比较模块可配置为产生边界对齐或中心对齐的脉冲波形。所有 PWM 操作都是双重缓冲的（缓冲寄存器在模块内部，并且不映射到 SFR 空间）。

为 PWM 操作设置模块：

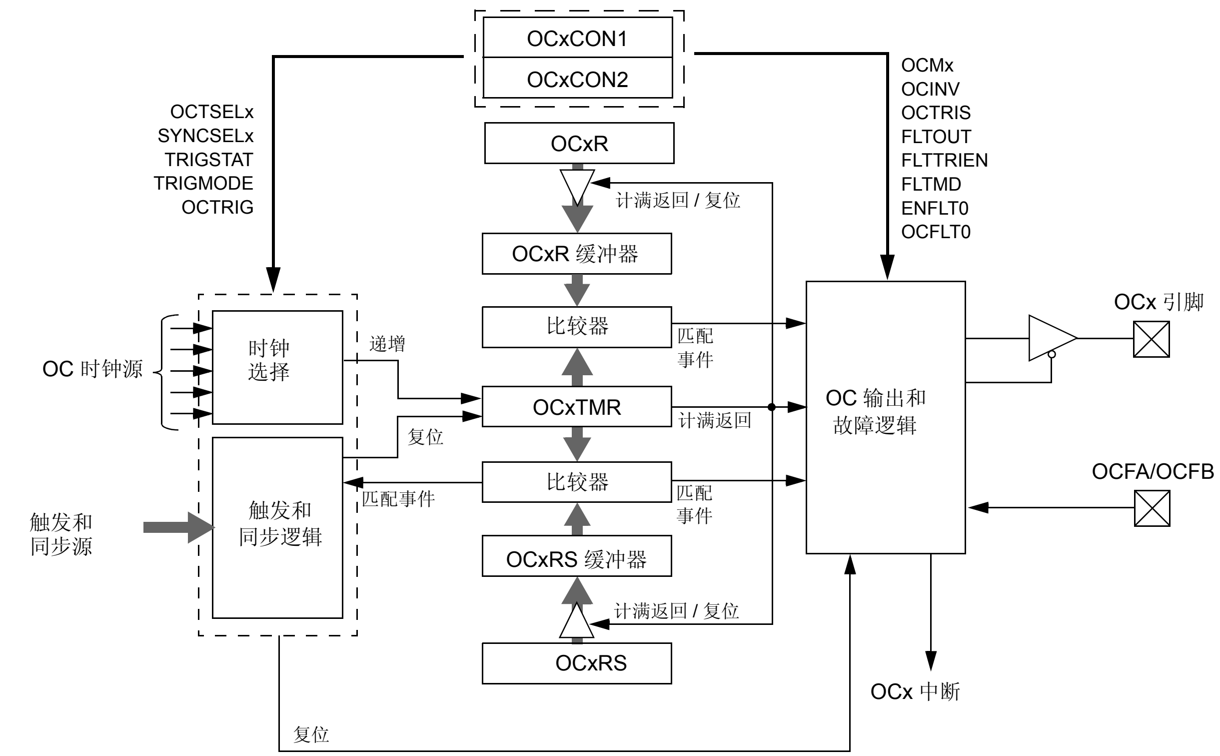
1. 将 OCx 输出配置为可用的外设引脚选择引脚之一。
2. 计算所要求的占空比并将其装入 OCxR 寄存器。

注：	OCxR 寄存器应该在输出比较模块第一次使能之前被初始化。当模块在 PWM 模式下工作时，OCxR寄存器成为只读占空比寄存器。OCxR 寄存器中保存的值成为第一个 PWM 周期的 PWM 占空比。直到时基周期匹配发生，输出比较辅助寄存器 OCxRS 的内容才会被传输到 OCxR。

3. 计算所要求的周期并将其装入OCxRS寄存器中。
4. 通过将 0x1F 写入 SYNCSEL<4:0>（OCxCON2<4:0>）来选择当前的 OCx 作为触发 / 同步源。
5. 通过对 OCTSEL2<2:0>（OCxCON<12:10>）位写入数据来选择时钟源。
6. 如果需要的话，允许定时器和输出比较模块中断。使用 PWM 故障引脚时需要输出比较中断。
7. 在 OCM<2:0>（OCxCON1<2:0>）位中选择所要求的 PWM 模式。
8. 如果选择定时器作为时钟源，则设置 TMRy 的预分频比并通过将 TON（TxCON<15>）位置 1 来使能时基。

注：	本外设包含的输入和输出功能可能需要通过外设引脚选择功能进行配置。欲知更多信息，请参见**第 9.4 节“外设引脚选择”**。

图 13-2： 输出比较框图 （双重缓冲的 16 位 PWM 模式）



注 1： 在使用之前， OCx 输出必须被分配给可用的 RPn 引脚。欲知更多信息，请参见**第 9.4 节 “外设引脚选择”**。

13.3.1 PWM 周期

PWM 周期可通过写入 PRy （定时器周期寄存器）来指定。可以使用公式 13-1 来计算 PWM 周期。

公式 13-1： 计算 PWM 周期 (1)

PWM 周期 = [(PRy) + 1] • TCY •(定时器预分频值)

其中： PWM 频率 = 1/[PWM 周期]

注 1： 基于 TCY ＝ TOSC * 2，禁止打盹模式和 PLL。

> 注： 若 PRy 的值为 N，则会使 PWM 周期为 N + 1 个时基计数周期。例如，写入 PRy 寄存器的值为 7 将产生由 8 个时基周期组成的 PWM 周期。

13.3.2 PWM 占空比

PWM 占空比是通过写 OCxRS 和 OCxR 寄存器来指定的。在任何时间都可以对 OCxRS 和 OCxR 寄存器进行写操作，但是在 PRy 和 TMRy 发生匹配（即周期完成）前占空比值不会被锁存。这可以为 PWM 占空比提供双重缓冲，对于 PWM 的无毛刺操作是极其重要的。

以下是 PWM 占空比的部分重要边界参数：

- 如果 OCxR、 OCxRS 和 PRy 都装载 0000h， OCx 引脚将保持低电平 （占空比为 0%）。
- 如果 OCxRS 的值大于 PRy 的值，则引脚将保持高电平 （占空比为 100%）。

欲知 PWM 模式时序的详细信息，请参见例 13-1。表 13-1 和表 13-2 给出了当器件工作在 4 个 MIPS 和 10 个 MIPS 时相应的 PWM 频率和分辨率的示例。

 超前信息

公式 13-2： 计算 PWM 最大分辨率 (1)

$$\text{PWM 最大分辨率（位）} = \frac{\log_{10}\left(\frac{F_{CY}}{F_{PWM} \cdot (\text{定时器预分频值})}\right)}{\log_{10}(2)} \text{ 位}$$

注 1：根据 FCY = FOSC/2，禁止打盹模式和 PLL。

例 13-1： 计算 PWM 周期和占空比 (1)

1. 当 FOSC= 8 MHz 且带 PLL 功能 （器件时钟速率为 32 MHz）且 Timer2 的预分频比设置为 1:1 时，计算对应于所需的 PWM 频率为 52.08 kHz 的定时器周期寄存器的值。

   TCY = 2 * TOSC = 62.5 ns

   PWM 周期 = 1/PWM 频率 = 1/52.08 kHz = 19.2 μs

   PWM 周期 = (PR2 + 1) • TCY • (Timer2 预分频值)

   19.2 μs = (PR2 + 1) • 62.5 ns •1

   PR2 = 306

2. 当 PWM 频率为 52.08 kHz 且器件时钟速率为 32 MHz 时，计算占空比的最大分辨率：

   PWM 分辨率 = $\log_{10}(F_{CY}/F_{PWM})/\log_{10}2)$ 位

   = $(\log_{10}(16\ \text{MHz}/52.08\ \text{kHz})/\log_{10}2)$ 位

   = 8.3 位

注 1： 基于 TCY = 2 * TOSC ；禁止打盹模式和 PLL。

表 13-1： 器件工作在 4 MIPS （FCY = 4 MHz）时的 PWM 频率和分辨率示例 (1)

PWM 频率	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
定时器预分频比	8	1	1	1	1	1	1
周期寄存器值	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分辨率 (位)	16	16	15	12	10	7	5

注 1： 基于 FCY = FOSC/2，禁止打盹模式和 PLL。

表 13-2： 器件工作在 16 MIPS （FCY = 16 MHz）时的 PWM 频率和分辨率示例 (1)

PWM 频率	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
定时器预分频比	8	1	1	1	1	1	1
周期寄存器值	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分辨率 (位)	16	16	15	12	10	7	5

注 1： 基于 FCY = FOSC/2，禁止打盹模式和 PLL。

寄存器 13-1： **OCxCON1：输出比较 x 控制 1 寄存器**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—
bit 15							bit 8

R/W-0	U-0	U-0	R/W-0, HCS	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2(1)	OCM1(1)	OCM0(1)
bit 7							bit 0

图注：	HCS = 可由硬件清零 / 置 1 的位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为 0

bit 13 **OCSIDL：**在空闲模式下停止输出比较 x 控制位
1 = 输出比较 x 在 CPU 空闲模式下暂停
0 = 输出比较 x 在 CPU 空闲模式下继续工作

bit 12-10 **OCTSEL2:OCTSEL0：**输出比较 x 定时器选择位
111 = 系统时钟
110 = 保留
101 = 保留
100 = Timer1
011 = Timer5
010 = Timer4
001 = Timer3
000 = Timer2

bit 9-8 **未实现：**读为 0

bit 7 **ENFLT0：**故障 0 输入使能位
1 = 故障 0 输入使能
0 = 故障 0 输入禁止

bit 6-5 **未实现：**读为 0

bit 4 **OCFLT0：**PWM 故障条件状态位
1 = 已经产生 PWM 故障条件 （仅由硬件清零）
0 = 没有产生 PWM 故障条件 （只有在 OCM<2:0>= 111 时才使用该位）

bit 3 **TRIGMODE：**触发状态模式选择位
1 = TRIGSTAT （OCxCON2<6>）在 OCxRS = OCxTMR 时或者在软件中清零
0 = TRIGSTAT 仅由软件清零

bit 2-0 **OCM2:OCM0：**输出比较 x 模式选择位 (1)
111 = 在 OCx 上输出中心对齐 PWM(2)
110 = 在 OCx 上输出边沿对齐 PWM(2)
101 = 双比较连续脉冲模式：初始化 OCx 引脚为低电平，在连续周期内 OCxR 与 OCxRS 接连匹配时，持续翻转 OCx 状态
100 = 双比较单脉冲模式：初始化 OCx 引脚为低电平，在一个周期内 OCxR 与 OCxRS 匹配时，翻转 OCx 状态
011 = 单比较连续脉冲模式：比较事件持续翻转 OCx 引脚
010 = 单比较单脉冲模式：初始化 OCx 引脚为高电平，比较事件强制 OCx 引脚为低电平
001 = 单比较单脉冲模式：初始化 OCx 引脚为低电平，比较事件强制 OCx 引脚为高电平
000 = 禁止输出比较通道

注 1： OCx 输出也必须配置给可用的 RPn 引脚。欲知更多信息，请参见**第 9.4 节 “外设引脚选择”**。
2： OCFA 引脚控制 OC1 至 OC4 通道； OCFB 引脚控制 OC5 至 OC9 通道。 OCxR 和 OCxRS 仅在 PWM 模式下是双重缓冲的。

寄存器 13-2： OCxCON2：输出比较 x 控制 2 寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32
bit 15							bit 8

R/W-0	R/W-0, HS	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

图注：	HS = 可由硬件置 1 的位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **FLTMD**：故障模式选择位
1 = 故障模式保持，直到消除故障源并且相应的 OCFLT0 位由软件清零
0 = 故障模式保持，直到消除故障源并且启动了新的 PWM 周期

bit 14 **FLTOUT**：故障输出位
1 = PWM 输出在发生故障事件时被驱动为高电平
0 = PWM 输出在发生故障事件时被驱动为低电平

bit 13 **FLTTRIEN**：故障输出状态选择位
1 = 引脚在发生故障条件时强制为输出状态
0 = 引脚 I/O 状态不受故障影响

bit 12 **OCINV**：OCMP 极性反转位
1 = OCx 输出的极性反转
0 = OCx 输出的极性不反转

bit 11-9 **未实现**：读为 0

bit 8 **OC32**：级联两个 OC 模块的使能位（32 位操作）
1 = 使能级联模块操作
0 = 禁止级联模块操作

bit 7 **OCTRIG**：OCx 触发 / 同步选择位
1 = OCx 由 SYNCSELx 位设定的触发源触发
0 = OCx 由 SYNCSELx 位设定的同步源同步

bit 6 **TRIGSTAT**：定时器触发状态位
1 = 定时器源已触发并正在运行
0 = 定时器源未触发并保持清零状态

bit 5 **OCTRIS**：OCx 输出引脚方向选择位
1 = OCx 引脚是三态引脚
0 = OCx 引脚上连接了输出比较外设 x

注 **1**：绝不要通过选择该模式或其他相当的 SYNCSEL 设置将 OC 模块用作其自身触发源。
2：仅将这些输入作为触发源，从不用作同步源。

寄存器 13-2： OCxCON2：输出比较 x 控制 2 寄存器 （续）

bit 4-0 **SYNCSEL4:SYNCSEL0：**触发 / 同步源选择位
11111 = 这个 OC 模块 (1)
11110 = 输入捕捉 9(2)
11101 = 输入捕捉 6(2)
11100 = CTMU(2)
11011 = A/D(2)
11010 = 比较器 3(2)
11001 = 比较器 2(2)
11000 = 比较器 1(2)
10111 = 输入捕捉 4(2)
10110 = 输入捕捉 3(2)
10101 = 输入捕捉 2(2)
10100 = 输入捕捉 1(2)
10011 = 输入捕捉 8(2)
10010 = 输入捕捉 7(2)
1000x = 保留
01111 = Timer 5
01110 = Timer 4
01101 = Timer 3
01100 = Timer 2
01011 = Timer 1
01010 = 输入捕捉 5(2)
01001 = 输出比较 9(1)
01000 = 输出比较 8(1)
00111 = 输出比较 7(1)
00110 = 输出比较 6(1)
00101 = 输出比较 5(1)
00100 = 输出比较 4(1)
00011 = 输出比较 3(1)
00010 = 输出比较 2(1)
00001 = 输出比较 1(1)
00000 = 不与任何其他模块同步

注 **1：** 绝不要通过选择该模式或其他相当的 SYNCSEL 设置将 OC 模块用作其自身触发源。
2： 仅将这些输入作为触发源，从不用作同步源。

 超前信息

14.0 串行外设接口（SPI）

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的“**第 23 章 串行外设接口（SPI）**”（DS39699A_CN）。

串行外设接口（Serial Peripheral Interface，SPI）模块是一个同步串行接口，可用于与其他外设或者单片机进行通信。这些外设器件可以是串行 EEPROM、移位寄存器、显示驱动器、A/D 转换器等。SPI 模块与 Motorola 的 SPI 和 SIOP 接口兼容。PIC24FJ256GA110 系列中全部的器件都包含三个 SPI 模块。

该模块可在两种缓冲模式下工作。在标准模式中，通过单个串行缓冲器移动数据。在增强型缓冲模式中，通过 8 级深 FIFO 缓冲器移动数据。

注：	无论是在标准还是增强型缓冲模式下，都不要对 SPIxBUF 寄存器执行读－修改－写操作（如位操作指令）。

当工作在主模式或从模式下时，该模块还支持基本帧 SPI 协议。支持全部四种帧 SPI 配置。

SPI 串行接口由以下四个引脚组成：

- SDIx：串行数据输入
- SDOx：串行数据输出
- SCKx：移位时钟输入或输出
- $\overline{\text{SSx}}$：低电平有效的从选择或帧同步 I/O 脉冲

SPI 模块可以被配置为使用 2 个、3 个或 4 个引脚工作。在 3 引脚模式下，不使用 SSx。在 2 引脚模式下，不使用 SDOx 和 SSx。

图 14-1 和图 14-2 所示为标准模式和增强型模式下该模块的框图。

注：	在这里，SPI 模块统称为 SPIx，或分别称为 SPI1、SPI2 或 SPI3。特殊功能寄存器也使用类似的符号表示。例如，SPIxCON1 和 SPIxCON2 指 3 个 SPI 模块中任一模块的控制寄存器。

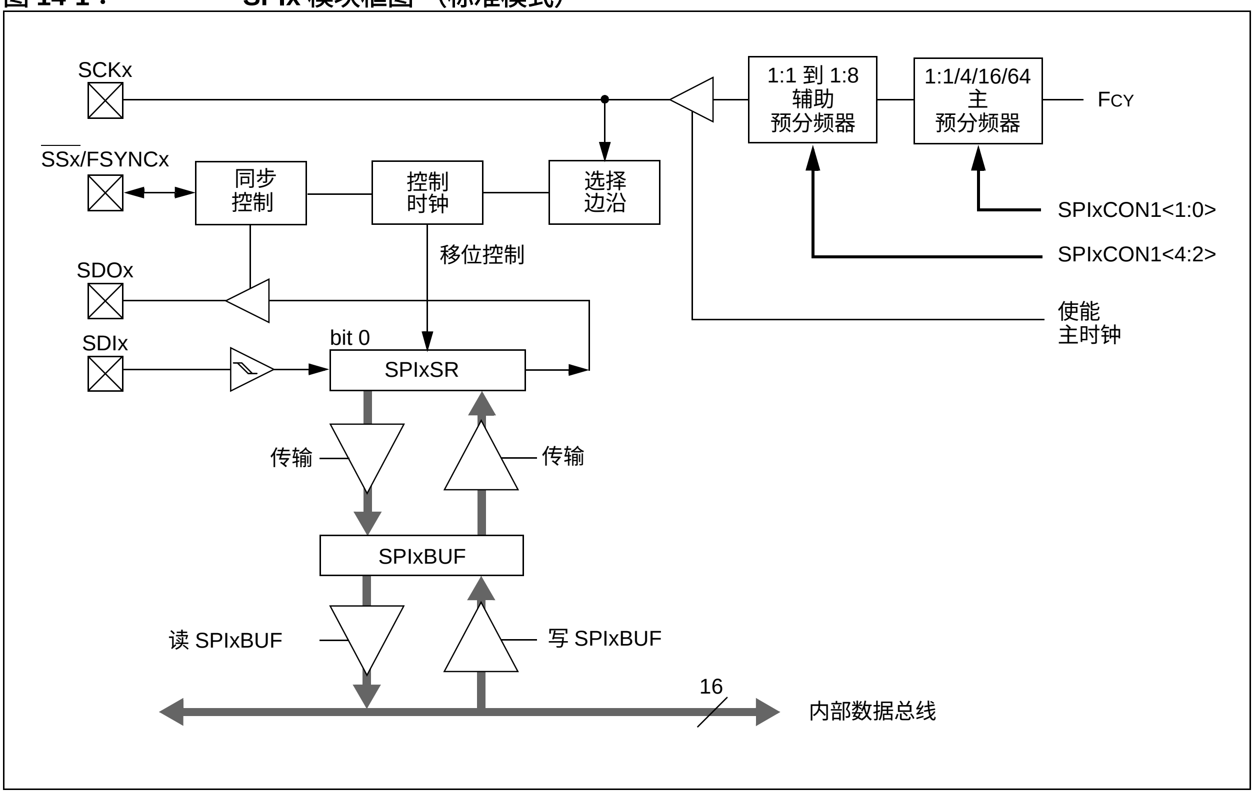
设置 SPI 模块工作在标准主模式下：

1. 如果使用中断：
   a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
   b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
   c) 通过写相应 IPC 寄存器中的 SPIxIP 位来设置中断优先级。
2. 将要求的设置写入 SPIxCON1 和 SPIxCON2 寄存器且 MSTEN 位 （SPIxCON1<5>）= 1。
3. 将 SPIROV 位 （SPIxSTAT<6>）清零。
4. 通过将 SPIEN 位 （SPIxSTAT<15>）置 1 使能 SPI。
5. 将待发送数据写入 SPIxBUF 寄存器。数据一写入 SPIxBUF 寄存器，发送 （和接收）就会立即开始。

设置 SPI 模块工作在标准从模式下：

1. 将 SPIxBUF 寄存器清零。
2. 如果使用中断：
   a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
   b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
   c) 通过写相应 IPC 寄存器中的 SPIxIP 位来设置中断优先级。
3. 将要求的设置写入 SPIxCON1 和 SPIxCON2 寄存器且 MSTEN 位 （SPIxCON1<5>）= 0。
4. 将 SMP 位清零。
5. 如果将 CKE 位置 1，然后必须将 SSEN 位 （SPIxCON1<8>）置 1 来使能 SSx 引脚。
6. 将 SPIROV 位 （SPIxSTAT<6>）清零。
7. 通过将 SPIEN 位 （SPIxSTAT<15>）置 1 使能 SPI。

图 14-1： SPIx 模块框图 （标准模式）



 超前信息

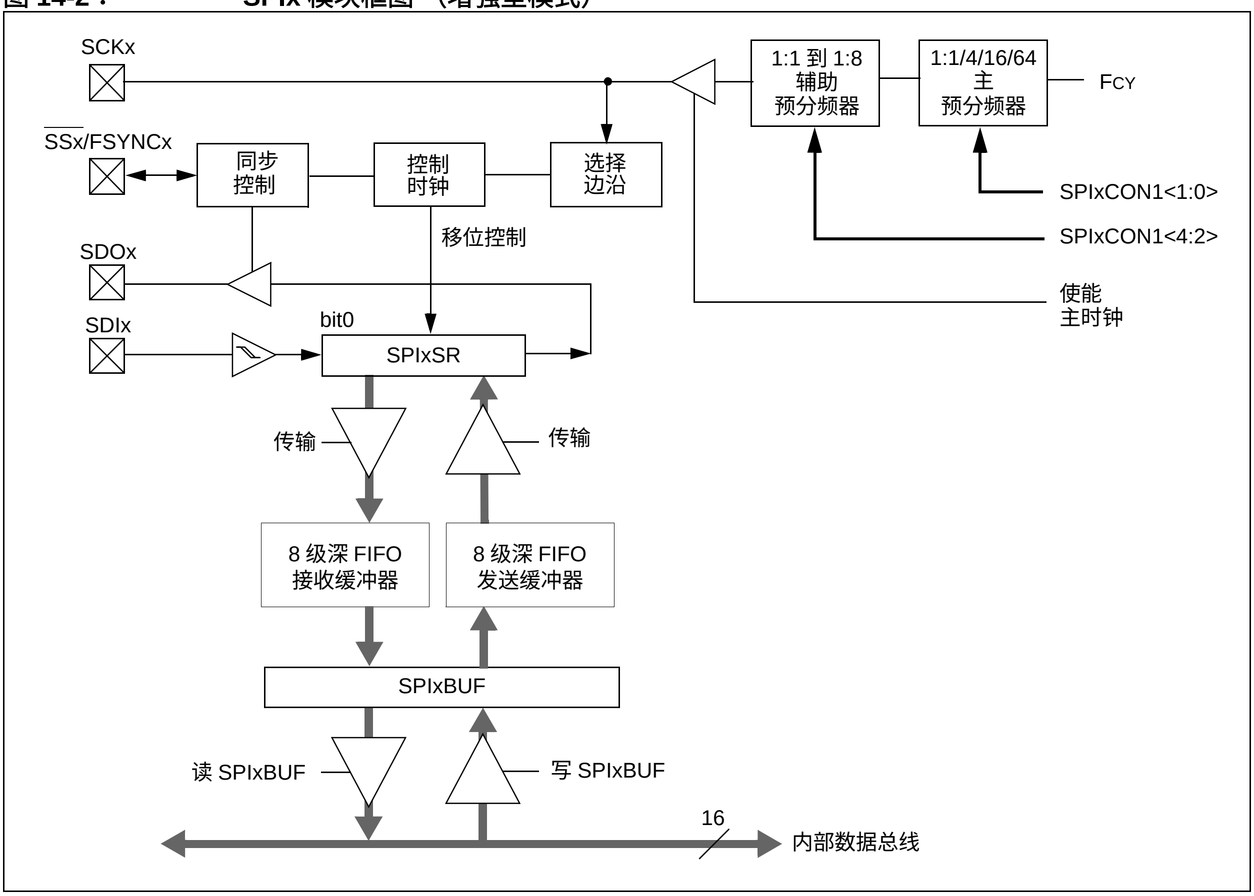
设置 SPI 模块工作在增强型缓冲器主模式下：

1. 如果使用中断：
   a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
   b) 将相应 IECx 寄存器中的 SPIxIE 位置 1。
   c) 写相应 IPC 寄存器中的 SPIxIP 位。
2. 将要求的设置写入 SPIxCON1 和 SPIxCON2 寄存器且 MSTEN 位 （SPIxCON1<5>）= 1。
3. 将 SPIROV 位 （SPIxSTAT<6>）清零。
4. 通过将 SPIBEN 位 （SPIxCON2<0>）置 1 选择增强型缓冲模式。
5. 通过将 SPIEN 位 （SPIxSTAT<15>）置 1 使能 SPI。
6. 将待发送数据写入 SPIxBUF 寄存器。数据一写入 SPIxBUF 寄存器，发送 （和接收）就会立即开始。

设置 SPI 模块工作在增强型缓冲从模式下：

1. 将 SPIxBUF 寄存器清零。
2. 如果使用中断：
   a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
   b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
   c) 通过写相应 IPC 寄存器中的 SPIxIP 位来设置中断优先级。
3. 将要求的设置写入 SPIxCON1 和 SPIxCON2 寄存器且 MSTEN 位 （SPIxCON1<5>）= 0。
4. 将 SMP 位清零。
5. 如果 CKE 位置 1，则 SSEN 位也必须置 1 以使能 $\overline{SSx}$ 引脚。
6. 将 SPIROV 位 （SPIxSTAT<6>）清零。
7. 通过将 SPIBEN 位 （SPIxCON2<0>）置 1 选择增强型缓冲模式。
8. 通过将 SPIEN 位 （SPIxSTAT<15>）置 1 使能 SPI。

图 14-2： SPIx 模块框图 （增强型模式）



寄存器 14-1：　SPIxSTAT：SPIx 状态和控制寄存器

R/W-0	U-0	R/W-0	U-0	U-0	R-0	R-0	R-0
SPIEN[1]	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0
bit 15							bit 8

R-0	R/C-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0
SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF
bit 7							bit 0

图注：	C = 可清零位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15　**SPIEN：**SPIx 使能位 [1]
　1 = 使能模块并将 SCKx、 SDOx、 SDIx 和 $\overline{SSx}$ 配置为串口引脚
　0 = 禁止模块

bit 14　**未实现：**读为 0

bit 13　**SPISIDL：**空闲模式停止位
　1 = 当器件进入空闲模式后，模块停止工作
　0 = 模块在空闲模式下继续工作

bit 12-11　**未实现：**读为 0

bit 10-8　**SPIBEC2:SPIBEC0：**SPIx 缓冲器元素计数位 （在增强型缓冲模式中有效）
　主模式：
　等待中的 SPI 传输数
　从模式：
　未读取的 SPI 传输数

bit 7　**SRMPT：**移位寄存器 （SPIxSR）空位 （在增强型缓冲模式中有效）
　1 = SPIx 移位寄存器为空并已做好发送或接收的准备
　0 = SPIx 移位寄存器非空

bit 6　**SPIROV：**接收溢出标志位
　1 = 一个新字节 / 字已接收完毕并被丢弃。用户软件尚未读取 SPIxBUF 寄存器中先前接收到的数据。
　0 = 未发生溢出

bit 5　**SRXMPT：**接收 FIFO 空位 （在增强型缓冲模式中有效）
　1 = 接收 FIFO 为空
　0 = 接收 FIFO 不为空

bit 4-2　**SISEL2:SISEL0：**SPIx 缓冲器中断模式位 （在增强型缓冲模式中有效）
　111 = 当 SPIx 发送缓冲器满时产生中断 （SPITBF 位置 1）
　110 = 最后一位移入 SPIxSR 导致发送 FIFO 为空时产生中断
　101 = 最后一位被移出 SPIxSR 时产生中断，现在发送完成
　100 = 有数据移入 SPIxSR 导致发送 FIFO 有待发送数据时产生中断
　011 = 当 SPIx 接收缓冲器满时产生中断 （SPIRBF 位置 1）
　010 = 当 SPIx 接收缓冲器填满 3/4 或更多时产生中断
　001 = 接收缓冲器中有数据时产生中断 （SRMPT 位置 1）
　000 = 读取接收缓冲器中的最后一个数据导致缓冲器为空时产生中断 （SRXMPT 位置 1）

注　1： 如果 SPIEN = 1，这些功能在使用之前必须分配给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

寄存器 14-1： SPIxSTAT：SPIx 状态和控制寄存器 （续）

bit 1 **SPITBF**：SPIx 发送缓冲器满状态位
1 = 发送未开始， SPIxTXB 已满
0 = 发送已开始， SPIxTXB 为空
在标准缓冲模式中：
当 CPU 通过写 SPIxBUF 地址单元装载 SPIxTXB 时，该位由硬件自动置 1。当 SPIx 模块将数据从 SPIxTXB 传输到 SPIxSR 时，该位由硬件自动清零。
在增强型缓冲模式中：
当 CPU 通过写 SPIxBUF 地址单元装载最后一个可用缓冲单元时，该位由硬件自动置 1。当有空的缓冲单元可由 CPU 写入时由硬件自动清零。

bit 0 **SPIRBF**：SPIx 接收缓冲器满状态位
1 = 接收完成， SPIxRXB 已满
0 = 接收未完成， SPIxRXB 为空
在标准缓冲模式中：
当 SPIx 将数据从 SPIxSR 传输到 SPIxRXB 时，该位由硬件自动置 1。当内核通过读 SPIxBUF 地址单元读 SPIxRXB 时，该位由硬件自动清零。
在增强型缓冲模式中：
当 SPIx 将数据从 SPIxSR 传输到缓冲器填充了最后一个未读的缓冲单元时，该位由硬件自动置 1。当有空的缓冲单元可接收来自 SPIxSR 的传输数据时，该位由硬件自动清零。

注 1： 如果 SPIEN = 1，这些功能在使用之前必须分配给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

寄存器 14-2： SPIxCON1：SPIx 控制寄存器 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK(1)	DISSDO(2)	MODE16	SMP	CKE(3)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEN(4)	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-13 **未实现：**读为 0

bit 12 **DISSCK：**禁止 SCKx 引脚位 （仅限 SPI 主模式） (1)
1 = 禁止内部 SPI 时钟；引脚作为 I/O 端口使用
0 = 使能内部 SPI 时钟

bit 11 **DISSDO：**禁止 SDOx 引脚位 (2)
1 = 模块不使用 SDOx 引脚；引脚作为 I/O 端口使用
0 = SDOx 引脚受模块控制

bit 10 **MODE16：**字 / 字节通信选择位
1 = 通信为字宽 （16 位）
0 = 通信为字节宽 （8 位）

bit 9 **SMP：**SPIx 数据输入采样阶段位
主模式：
1 = 在数据输出时间的末尾采样输入数据
0 = 在数据输出时间的中间采样输入数据
从模式：
当在从模式下使用 SPIx 时，必须将 SMP 清零。

bit 8 **CKE：**SPIx 时钟边沿选择位 (3)
1 = 串行输出数据在时钟由有效状态变为空闲状态时改变 （见 bit 6）
0 = 串行输出数据在时钟由空闲状态变为有效状态时改变 （见 bit 6）

bit 7 **SSEN：**从选择使能 （从模式）位 (4)
1 = $\overline{SSx}$ 引脚用于从模式
0 = 模块不使用$\overline{SSx}$ 引脚，引脚由端口功能控制

bit 6 **CKP：**时钟极性选择位
1 = 空闲状态时钟信号为高电平；有效状态为低电平
0 = 空闲状态时钟信号为低电平；有效状态为高电平

bit 5 **MSTEN：**主模式使能位
1 = 主模式
0 = 从模式

注 1： 如果 DISSCK = 0，必须将 SCKx 配置给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。
2： 如果 DISSDO= 0，必须将 SDOx 配置给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。
3： 在帧 SPI 模式下不使用 CKE 位。在帧 SPI 模式 （FRMEN = 1）下，用户应该将该位编程为 0。
4： 如果 SSEN = 1，必须将 $\overline{SSx}$ 配置给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

寄存器 14-2： SPIxCON1：SPIx 控制寄存器 1 （续）

bit 4-2 **SPRE2:SPRE0**：辅助预分频比位 （主模式）
111 = 辅助预分频比为 1:1
110 = 辅助预分频比为 2:1
...
000 = 辅助预分频比 8:1

bit 1-0 **PPRE1:PPRE0**：主预分频比位 （主模式）
11 = 主预分频比为 1:1
10 = 主预分频比为 4:1
01 = 主预分频比为 16:1
00 = 主预分频比为 64:1

注 1： 如果 DISSCK = 0，必须将 SCKx 配置给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。
2： 如果 DISSDO= 0，必须将 SDOx 配置给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。
3： 在帧 SPI 模式下不使用 CKE 位。在帧 SPI 模式 （FRMEN = 1）下，用户应该将该位编程为 0。
4： 如果 SSEN = 1，必须将 $\overline{SSx}$ 配置给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

寄存器 14-3： SPIxCON2：SPIx 控制寄存器 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SPIFE	SPIBEN
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **FRMEN**：帧 SPIx 支持位
1 = 使能帧 SPIx 支持
0 = 禁止帧 SPIx 支持

bit 14 **SPIFSD**：$\overline{SSx}$ 引脚上的帧同步脉冲方向控制位
1 = 帧同步脉冲输入 （从模式）
0 = 帧同步脉冲输出 （主模式）

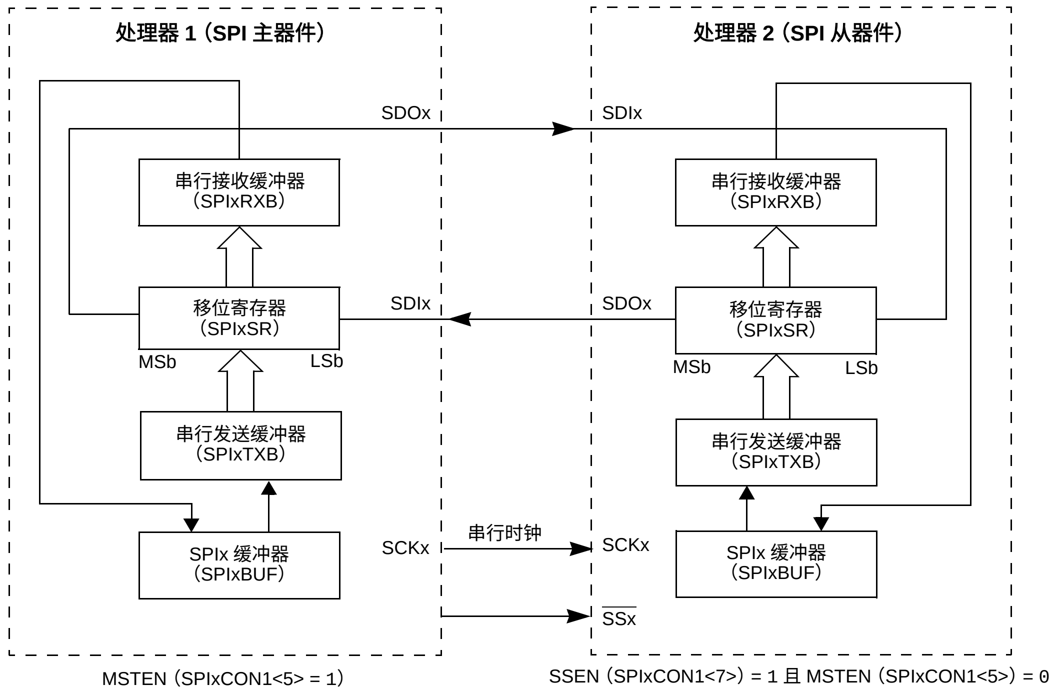
bit 13 **SPIFPOL**：帧同步脉冲极性位 （仅用于帧模式）
1 = 帧同步脉冲高电平有效
0 = 帧同步脉冲低电平有效

bit 12-2 **未实现**：读为 0

bit 1 **SPIFE**：帧同步脉冲边沿选择位
1 = 帧同步脉冲与第一个位时钟同步
0 = 帧同步脉冲比第一个位时钟超前

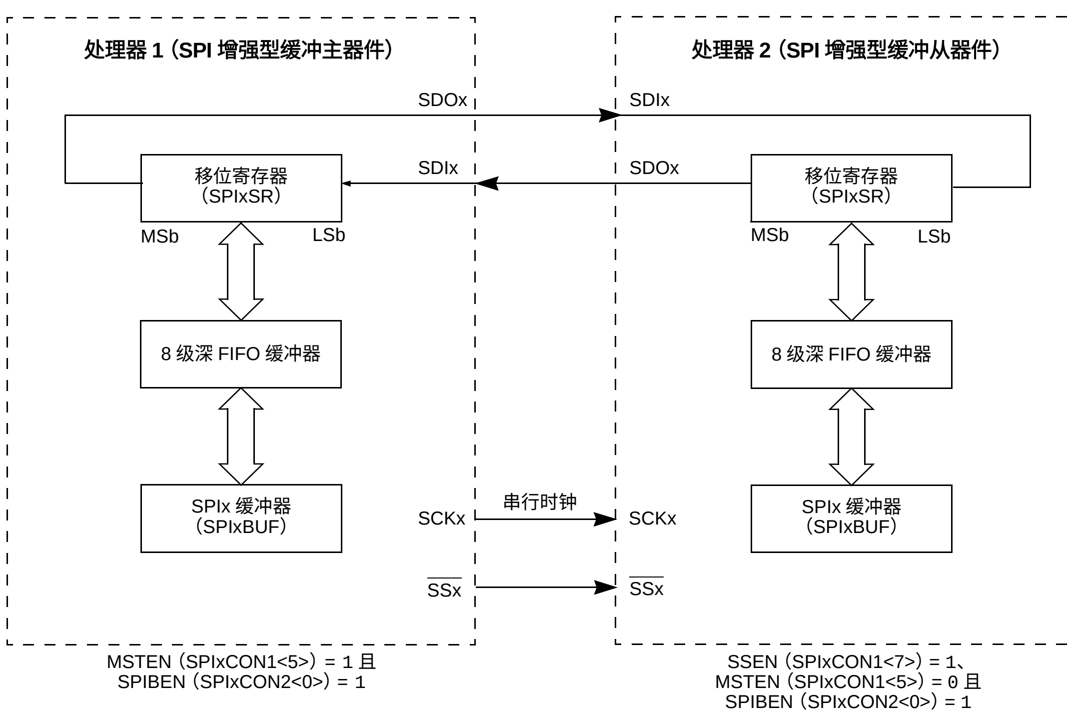
bit 0 **SPIBEN**：增强型缓冲器使能位
1 = 使能增强型缓冲器
0 = 禁止增强型缓冲器 （传统模式）

图 14-3： SPI 主 / 从连接 （标准模式）



注 1： 可选择在从模式下使用 $\overline{SSx}$ 引脚。
2： 用户必须向 SPIxBUF 写入发送数据，从 SPIxBUF 读取接收数据。 SPIxTXB 和 SPIxRXB 寄存器被存储器映射到 SPIxBUF。

图 14-4： SPI 主 / 从连接 （增强型缓冲模式）



注 1： 可选择在从模式下使用 $\overline{SSx}$ 引脚。
2： 用户必须向 SPIxBUF 写入发送数据，从 SPIxBUF 读取接收数据。SPIxTXB 和 SPIxRXB 寄存器被存储器映射到 SPIxBUF。

图 14-5： SPI 主器件、帧主器件连接图

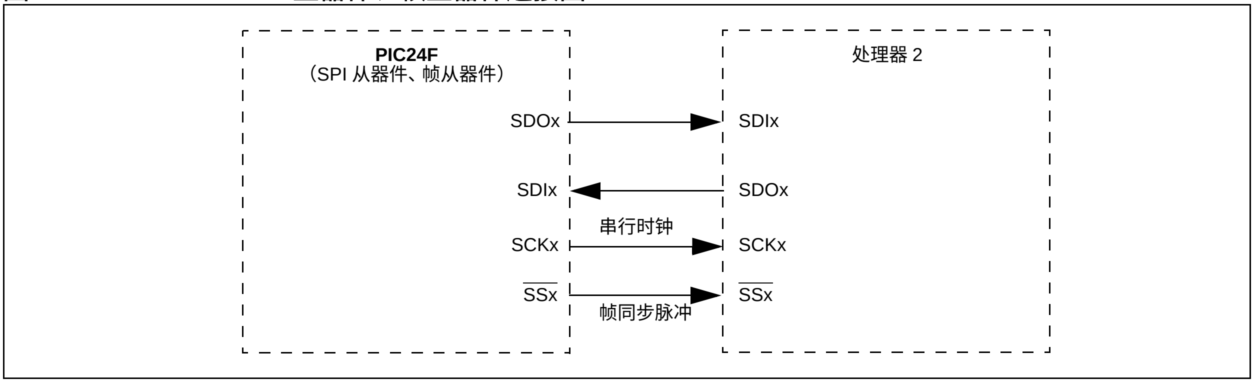


图 14-6： SPI 主器件、帧从器件连接图

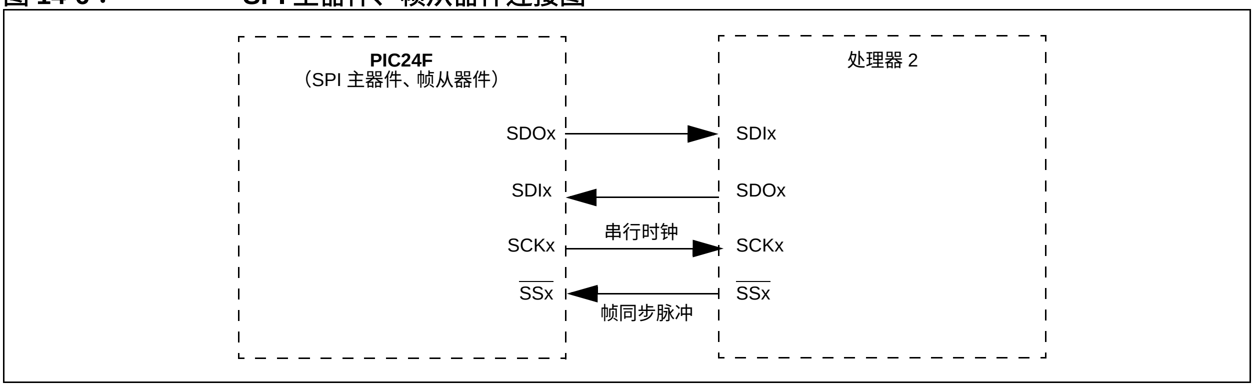


图 14-7： SPI 从器件、帧主器件连接图

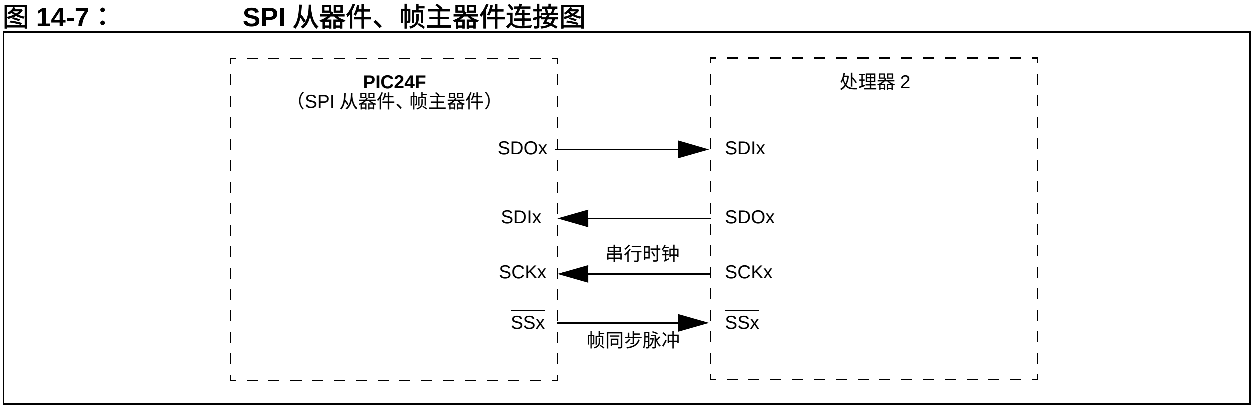
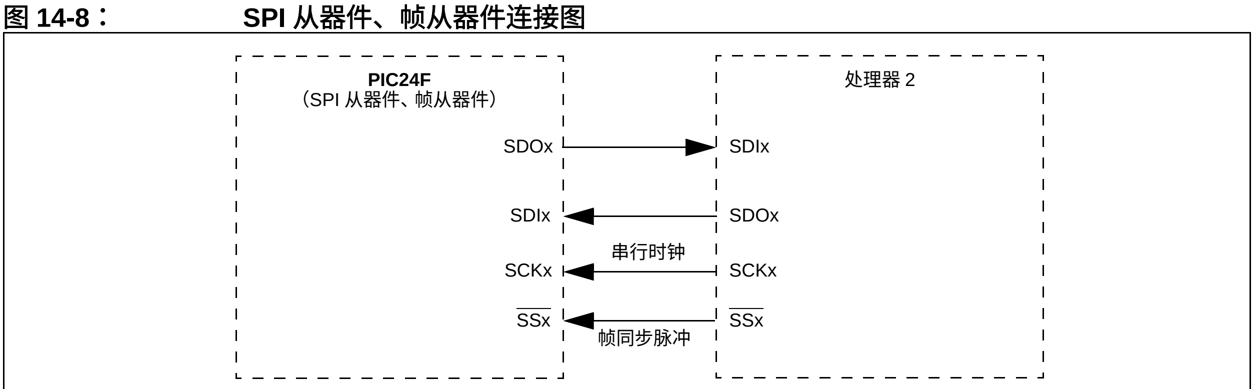


图 14-8： SPI 从器件、帧从器件连接图



公式 14-1： 器件速率和 SPI 时钟速度之间的关系 [1]

$$F_{SCK} = \frac{F_{CY}}{\text{主预分频比} * \text{辅助预分频比}}$$

注 1：基于 $F_{CY} = F_{OSC}/2$，禁止打盹模式和 PLL。

表 14-1： 采样 SCK 频率示例 [1,2]

FCY = 16 MHz		辅助预分频比设置				
		1:1	**2:1**	**4:1**	**6:1**	**8:1**
主预分频比设置	1:1	无效	8000	4000	2667	2000
	4:1	4000	2000	1000	667	500
	16:1	1000	500	250	167	125
	64:1	250	125	63	42	31
FCY = 5 MHz						
主预分频比设置	1:1	5000	2500	1250	833	625
	4:1	1250	625	313	208	156
	16:1	313	156	78	52	39
	64:1	78	39	20	13	10

注 1： 基于 $F_{CY} = F_{OSC}/2$，禁止打盹模式和 PLL。
2： SCKx 频率的单位为 kHz。

15.0 I²C™

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的 **“第 24 章 I²C™”**（DS39702A_CN）。

I²C 模块是用来与其他外设或单片机器件通信的串行接口。这些外设可以是串行 EEPROM、显示驱动器和 A/D 转换器等。

I²C 模块支持以下特性：

- 独立的主和从逻辑
- 7 位和 10 位器件地址
- I²C 协议定义的广播呼叫地址
- 时钟延长可为处理器提供响应从器件数据请求的延时
- 支持 100 kHz 和 400 kHz 两种总线规范
- 可配置的地址掩码
- 多主器件模式以防止仲裁时报文丢失
- 总线转发器模式，允许作为从器件验收所有报文而与地址无关
- 自动 SCL

模块的框图如图 15-1 所示。

15.1 外设重映射选项

I²C 模块的引脚分配是固定的，它不能通过外设引脚选择备用引脚。要想灵活地复用外设，可以在配置器件时将100引脚器件中的I²C2模块重新分配给指定为ASCL2 和 ASDA2 的备用引脚。

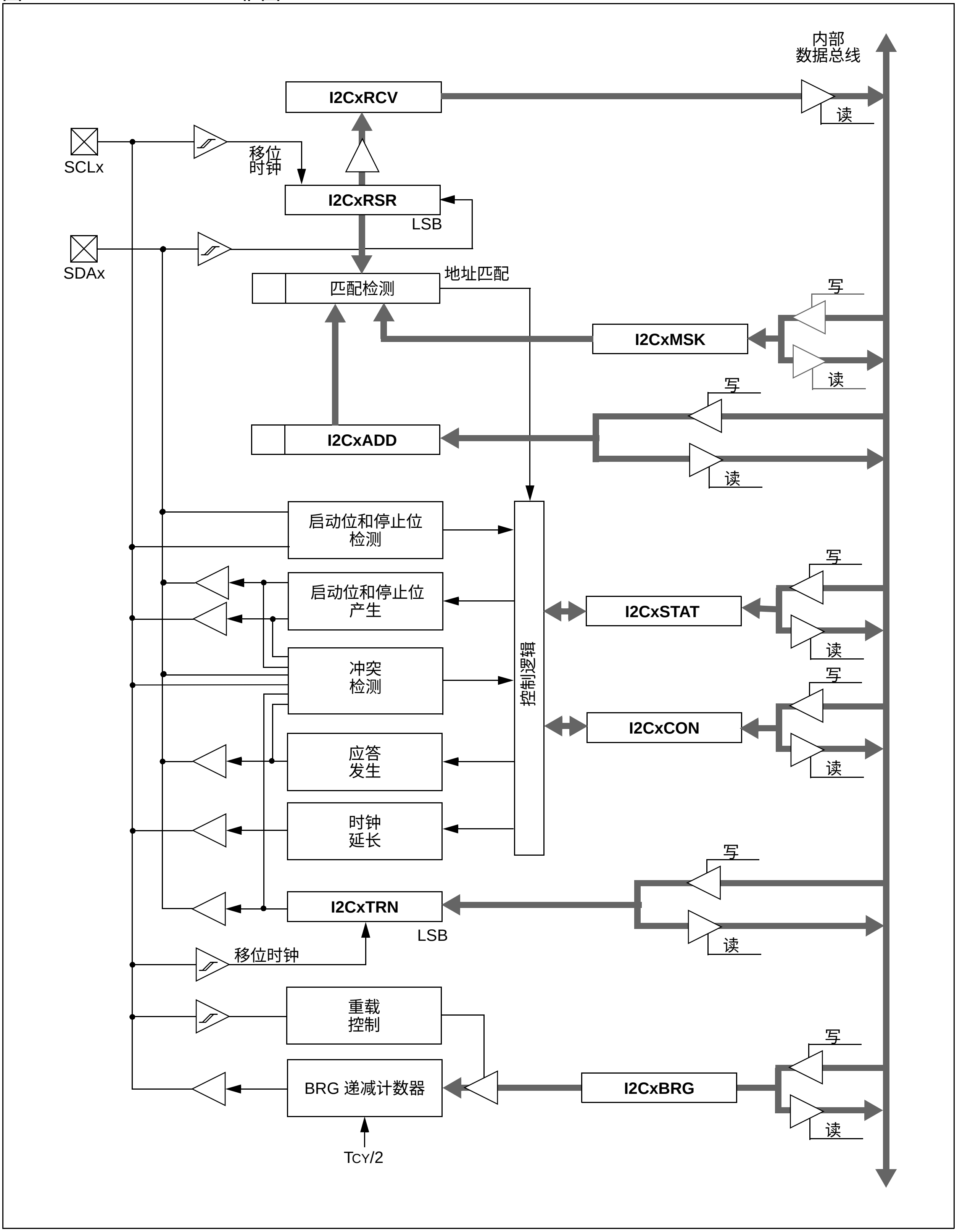
引脚的分配由I2C2SEL配置位控制；将该位编程为 `0`，使 ASCL2 和 ASDA2 引脚复用该模块。

15.2 作为主器件在单主器件环境中通信

在主模式下发送报文的细节取决于要与主器件进行通信的器件的通信协议。通常，发送报文的步骤如下：

1. 在 SDAx 和 SCLx 上发出一个 “ 启动 ” 条件。
2. 发送一个带写指示的 I²C 器件地址字节到从器件。
3. 等待并验证从器件的应答。
4. 发送第一个数据字节（有时被称为命令）到从器件。
5. 等待并验证从器件的应答。
6. 发送串行存储器地址低字节到从器件。
7. 重复步骤 4 和 5，直至所有数据字节发送结束。
8. 在 SDAx 和 SCLx 上发送一个“ 重复启动 ”条件。
9. 发送一个带读指示的器件地址字节到从器件。
10. 等待并验证从器件的应答。
11. 使能主器件接收以接收串行存储器数据。
12. 在数据字节接收结束时，产生 ACK 或 NACK 条件。
13. 在 SDAx 和 SCLx 上产生一个 “ 停止 ” 条件。

图 15-1：　　I²C™ 框图



　　超前信息　　

15.3 设置用作总线主器件时的波特率

使用公式 15-1 计算波特率发生器的重载值。

公式 15-1： 计算波特率重载值 (1,2)

$$F_{SCL} = \frac{F_{CY}}{I2CxBRG + 1 + \frac{F_{CY}}{10,000,000}}$$

或

$$I2CxBRG = \left(\frac{F_{CY}}{F_{SCL}} - \frac{F_{CY}}{10,000,000}\right) - 1$$

注 1： 基于 FCY = FOSC/2，禁止打盹模式和 PLL。

2： 这些时钟频率仅供参考。实际的时钟频率由多个系统级参数决定。应该在目标应用中测量实际时钟频率。

15.4 从地址掩码

I2CxMSK 寄存器 （见寄存器 15-3）指定了 7 位和 10 位地址模式下某些 “ 无关 ”地址位的位置。将 I2CxMSK 寄存器中某个特定位置 1 （= 1），不论相应的地址位是 0还是1，从模块都会做出响应。例如，当将I2CxMSK设置为 0010000 时，从模块将检测两个地址 0000000 和 0010000。

为了使能地址掩码，必须通过将 IPMIEN 位 （I2CxCON<11>）清零来禁止智能外设管理接口 （Intelligent Peripheral Management Interface，IPMI）。

注： 由于 I²C™ 协议被修改，表 15-2 中的地址将被保留并不会在从模式中被应答。这包括包含任何这些地址的任何地址掩码设置。

表 15-1： I²C™ 时钟速率 (1,2)

需要的系统 FSCL	FCY	I2CxBRG 值		实际 FSCL
		（十进制）	（十六进制）	
100 kHz	16 MHz	157	9D	100 kHz
100 kHz	8 MHz	78	4E	100 kHz
100 kHz	4 MHz	39	27	99 kHz
400 kHz	16 MHz	37	25	404 kHz
400 kHz	8 MHz	18	12	404 kHz
400 kHz	4 MHz	9	9	385 kHz
400 kHz	2 MHz	4	4	385 kHz
1 MHz	16 MHz	13	D	1.026 MHz
1 MHz	8 MHz	6	6	1.026 MHz
1 MHz	4 MHz	3	3	0.909 MHz

注 1： 基于 FCY = FOSC/2，禁止打盹模式和 PLL。

2： 这些时钟频率仅供参考。实际的时钟频率由多个系统级参数决定。应该在目标应用中测量实际时钟频率。

表 15-2： I²C™ 保留地址 (1)

从地址	R/$\overline{W}$ 位	说明
0000 000	0	广播呼叫地址 (2)
0000 000	1	启动字节
0000 001	x	CBus 地址
0000 010	x	保留
0000 011	x	保留
0000 1xx	x	HS 模式主代码
1111 1xx	x	保留
1111 0xx	x	10 位从地址高字节 (3)

注 1： 上述地址位永远不会导致地址匹配，无论地址掩码的设置如何。

2： 仅当 GCEN = 1 时，才会应答地址。

3： 与这些地址的匹配只会发生在 10 位寻址模式下的高字节中。

寄存器 15-1： I2CxCON：I2Cx 控制寄存器

R/W-0	U-0	R/W-0	R/W-1 HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

图注：	HC = 可由硬件清零的位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **I2CEN：**I2Cx 使能位
1 = 使能 I2Cx 模块并将 SDAx 和 SCLx 引脚配置为串口引脚
0 = 禁止 I2Cx 模块。所有 I^2C 引脚由端口功能控制。

bit 14 **未实现：**读为 0

bit 13 **I2CSIDL：**空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 模块在空闲模式下继续工作

bit 12 **SCLREL：**SCLx 释放控制位 （作为 I^2C 从器件工作时）
1 = 释放 SCLx 时钟
0 = 保持 SCLx 时钟为低电平 （时钟延长）
如果 STREN = 1：
该位可读可写 （即软件可写入 0 来启动时钟延长或写入 1 来释放时钟）。
在从器件发送开始时由硬件清零。
在从器件接收结束时由硬件清零。
如果 STREN = 0：
该位可读且可被置 1 （即软件只能写入 1 来释放时钟）。
在从器件发送开始时由硬件清零。

bit 11 **IPMIEN：**智能外设管理接口 （IPMI）使能位
1 = 使能 IPMI 支持模式；应答所有地址
0 = 禁止 IPMI 模式

bit 10 **A10M：**10 位从器件寻址位
1 = I2CxADD 是一个 10 位从地址
0 = I2CxADD 是一个 7 位从地址

bit 9 **DISSLW：**禁止斜率控制位
1 = 禁止斜率控制
0 = 使能斜率控制

bit 8 **SMEN：**SMBus 输入电平位
1 = 使能符合 SMBus 规范的 I/O 引脚门限值
0 = 禁止 SMBus 输入门限值

bit 7 **GCEN：**广播呼叫使能位 （作为 I^2C 从器件工作时）
1 = 允许在 I2CxRSR 接收到广播呼叫地址时产生中断
（已使能模块接收）
0 = 禁止广播呼叫地址

bit 6 **STREN：**SCLx 时钟延长使能位 （作为 I^2C 从器件工作时）
与 SCLREL 位配合使用。
1 = 使能软件或接收时钟延长
0 = 禁止软件或接收时钟延长

寄存器 15-1： I2CxCON：I2Cx 控制寄存器 （续）

bit 5 **ACKDT：**应答数据位 （作为 I²C 主器件工作时，适用于主器件接收操作。）
当软件启动应答序列时将发送的值。
1 = 在应答时发送 NACK
0 = 在应答时发送 ACK

bit 4 **ACKEN：**应答序列使能位 （作为 I²C 主器件工作时，适用于主器件接收操作。）
1 = 在SDAx和SCLx引脚上发出应答序列并发送ACKDT数据位。在主器件应答序列结束时由硬件清零。
0 = 应答序列不在进行中

bit 3 **RCEN：**接收使能位 （作为 I²C 从器件工作时）
1 = 使能 I²C 接收模式。在主器件接收完数据字节的八位时由硬件清零。
0 = 接收序列不在进行中

bit 2 **PEN：**停止条件使能位 （作为 I²C 主器件工作时）
1 = 在 SDAx 和 SCLx 引脚上发出停止条件。在主器件停止序列结束时由硬件清零。
0 = 停止条件不在进行中

bit 1 **RSEN：**重复启动条件使能位 （作为 I²C 主器件工作时）
1 = 在 SDAx 和 SCLx 引脚上发出重复启动条件。在主器件重复启动序列结束时由硬件清零。
0 = 重复启动条件不在进行中

bit 0 **SEN：**启动条件使能位 （作为 I²C 主器件工作时）
1 = 在 SDAx 和 SCLx 引脚上发出启动条件。主器件启动序列结束时由硬件清零。
0 = 启动条件不在进行中

寄存器 15-2： I2CxSTAT：I2Cx 状态寄存器

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
bit 15							bit 8

R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	RBF	TBF
bit 7							bit 0

图注：	C = 可清零位	HS = 可由硬件置 1 的位	HSC = 可由硬件置 1/ 清零的位
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **ACKSTAT：**应答状态位
1 = 上次检测到 NACK
0 = 上次检测到 ACK
应答结束时由硬件置 1 或清零。

bit 14 **TRSTAT：**发送状态位 （作为 I^2C 主器件工作时，适用于主器件发送操作。）
1 = 主器件正在发送 （8 位 +ACK）
0 = 主器件不在进行发送
在主器件发送开始时由硬件置 1。在从器件应答结束时由硬件清零。

bit 13-11 **未实现：**读为 0

bit 10 **BCL：**主器件总线冲突检测位
1 = 主器件工作期间检测到了总线冲突
0 = 未发生冲突
检测到总线冲突时由硬件置 1。

bit 9 **GCSTAT：**广播呼叫状态位
1 = 接收到了广播呼叫地址
0 = 未接收到广播呼叫地址
当地址与广播呼叫地址匹配时由硬件置 1。当检测到停止条件时由硬件清零。

bit 8 **ADD10：**10 位地址状态位
1 = 10 位地址匹配
0 = 10 位地址不匹配
当与匹配的 10 位地址的第 2 个字节匹配时由硬件置 1。当检测到停止条件时由硬件清零。

bit 7 **IWCOL：**写冲突检测位
1 = 因为 I^2C 模块忙，尝试写 I2CxTRN 寄存器失败
0 = 未发生冲突
当总线忙时写 I2CxTRN 会使硬件将该位置 1 （由软件清零）。

bit 6 **I2COV：**接收溢出标志位
1 = 当 I2CxRCV 寄存器仍然保存原先的字节时接收到了新字节
0 = 未溢出
尝试将数据从 I2CxRSR 传输到 I2CxRCV 时由硬件置 1 （由软件清零）。

bit 5 **D/$\overline{\text{A}}$：**数据 / $\overline{\text{地址}}$位 （作为 I^2C 从器件工作时）
1 = 表示上一次接收的字节是数据
0 = 表示上一次接收的字节是器件地址
器件地址匹配时由硬件清零。写入 I2CxTRN 或接收从器件字节时由硬件置 1。

寄存器 15-2： I2CxSTAT：I2Cx 状态寄存器 （续）

bit 4	**P**：停止位 1 = 表示上次检测到停止位 0 = 表示上一次未检测到停止位 当检测到启动、重复启动或停止条件时由硬件置 1 或清零。
bit 3	**S**：启动位 1 = 表示上一次检测到启动 （或重复启动）位 0 = 表示上一次未检测到启动位 当检测到启动、重复启动或停止条件时由硬件置 1 或清零。
bit 2	**R/$\overline{\text{W}}$**：读 / $\overline{\text{写}}$信息位 （作为 I^2C 从器件工作时） 1 = 读——表示数据自从器件输出 0 = 写——表示数据输入到从器件 接收到 I^2C 器件地址字节后由硬件置 1 或清零。
bit 1	**RBF**：接收缓冲器满状态位 1 = 接收完成， I2CxRCV 为满 0 = 接收未完成， I2CxRCV 为空 用接收到的字节写 I2CxRCV 时由硬件置 1。当用软件读 I2CxRCV 时由硬件清零。
bit 0	**TBF**：发送缓冲器满状态位 1 = 正在发送， I2CxTRN 为满 0 = 发送完成， I2CxTRN 为空 用软件写 I2CxTRN 时由硬件置 1。数据发送完成时由硬件清零。

寄存器 15-3： I2CxMSK：I2Cx 从模式地址掩码寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AMSK9	AMSK8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AMSK7	AMSK6	AMSK5	AMSK4	AMSK3	AMSK2	AMSK1	AMSK0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-10 **未实现：**读为 0

bit 9-0 **AMSK9:AMSK0：**地址中 bit x 的掩码选择位
1 = 使能输入报文的地址中 bit x 的掩码；在此位置上不需要位匹配
0 = 禁止 bit x 的掩码；此位需要位匹配

16.0 通用异步收发器 （UART）

> **注：** 本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的 **“ 第 21 章 UART ”**（DS39708A_CN）。

通用异步收发器 （Universal Synchronous Receiver Transmitter，UART）模块是 PIC24F 系列器件提供的串行 I/O模块之一。UART 是可以和外设（例如个人电脑、LIN、RS-232 和 RS-485 接口）通信的全双工异步系统。UART 模块还通过 $\overline{\text{UxCTS}}$ 和 $\overline{\text{UxRTS}}$ 引脚支持硬件流控制选项，并包括了 IrDA® 编码器和解码器。

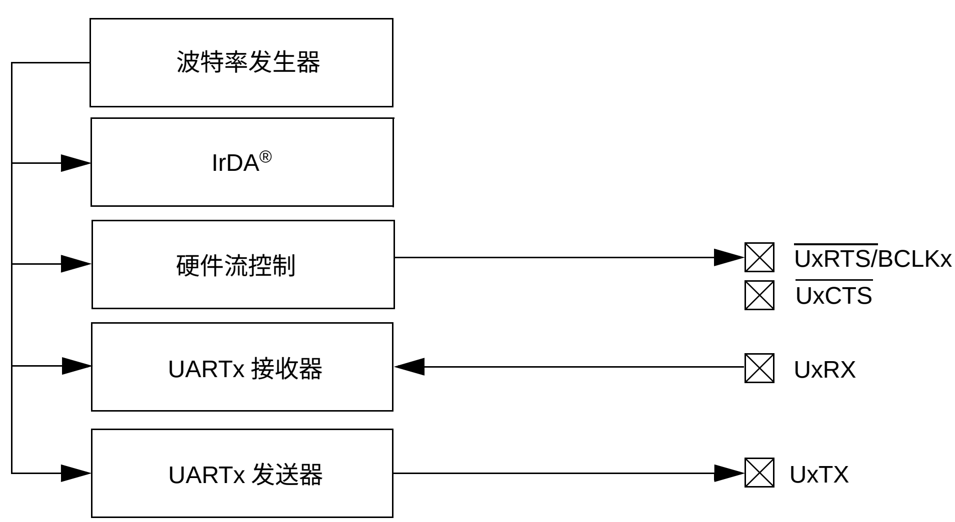
UART 模块的主要特性有：

- 通过 UxTX 和 UxRX 引脚进行全双工 8 位或 9 位数据传输
- 偶校验、奇校验或无奇偶校验选项 （针对 8 位数据）
- 一或两个停止位
- 通过 $\overline{\text{UxCTS}}$ 和 $\overline{\text{UxRTS}}$ 引脚支持的硬件流控制选项
- 完全集成的具有 16 位预分频器的波特率发生器
- 在 16 MIPS 下，波特率范围为 1 Mbps 到 15 Mbps
- 4 级深先进先出 （First-In-First-Out， FIFO）发送数据缓冲器
- 4 级深 FIFO 接收数据缓冲器
- 奇偶校验、帧和缓冲器溢出错误检测
- 支持带地址检测的 9 位模式（第 9 位 = 1）
- 发送和接收中断
- 支持诊断的环回模式
- 支持同步和间隔字符
- 支持自动波特率检测
- IrDA 编码器和解码器逻辑
- 支持 IrDA 的 16 倍频波特率时钟输出

图 16-1 所示为 UART 的简化框图。 UART 模块由以下关键硬件组成：

- 波特率发生器
- 异步发送器
- 异步接收器

图 16-1： UART 简化框图



注： 在使用前， UART 输入和输出必须分配给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

16.1 UART 波特率发生器 （BRG）

UART模块包含一个专用的16位波特率发生器。UxBRG寄存器控制一个自由运行的 16 位定时器的周期。公式16-1 给出了 BRGH = 0 时计算波特率的公式。

公式 16-1： BRGH = 0 时的 UART 波特率 [1,2]

$$\text{波特率} = \frac{F_{CY}}{16 \cdot (UxBRG + 1)}$$

$$UxBRG = \frac{F_{CY}}{16 \cdot \text{波特率}} - 1$$

注 1： FCY 表示指令周期的时钟频率（FOSC/2）。
2： 基于 FCY = FOSC/2，禁止打盹模式和 PLL。

例 16-1 所示为下列条件下波特率误差的计算：

- FCY = 4 MHz
- 目标波特率 = 9600

可能的最大波特率 （当 BRGH = 0 时）是 FCY/16 （当 UxBRG = 0 时），可能的最小波特率是 FCY/(16 * 65536)。

公式 16-2 给出了 BRGH = 1 时计算波特率的公式。

公式 16-2： BRGH = 1 时的 UART 波特率 [1,2]

$$\text{波特率} = \frac{F_{CY}}{4 \cdot (UxBRG + 1)}$$

$$UxBRG = \frac{F_{CY}}{4 \cdot \text{波特率}} - 1$$

注 1： FCY 表示指令周期的时钟频率。
2： 基于 FCY = FOSC/2，禁止打盹模式和 PLL。

可能的最大波特率（BRGH = 1时）是FCY/4（当UxBRG = 0时），可能的最小波特率是 FCY/(4 * 65536)。

向 UxBRG 寄存器中写入新值会导致 BRG 定时器复位（清零）。这保证了 BRG 在新的波特率产生之前不需等待定时器溢出。

例 16-1： 波特率误差计算 （当 BRGH = 0 时） [1]

目标波特率 = FCY/(16 (UxBRG + 1))

UxBRG 值的计算方法：

UxBRG = ((FCY/ 目标波特率)/16) – 1
UxBRG = ((4000000/9600)/16) – 1
UxBRG = 25

计算波特率 = 4000000/(16 (25 + 1))
= 9615

$$\text{误差} = \frac{(\text{计算波特率} - \text{目标波特率})}{\text{目标波特率}}$$
= (9615 – 9600)/9600
= 0.16%

注 1： 基于 FCY = FOSC/2，禁止打盹模式和 PLL。

16.2 在 8 位数据模式下发送

1. 设置 UART：
   a) 将适当的值写入数据、奇偶校验和停止位。
   b) 向 UxBRG 寄存器写入适当波特率值。
   c) 设置发送和接收的中断允许和优先级位。
2. 使能 UART。
3. 将 UTXEN 位置 1 （将该位置 1 后两个周期产生发送中断）。
4. 向 UxTXREG 字的低字节写入数据。该数据字节将立即传输给发送移位寄存器 （Transmit Shift Register， TSR）且在波特时钟的下一个上升沿开始移出串行比特流。
5. 也可以在 UTXEN = 0 时传输数据字节，然后用户将 UTXEN 置 1。因为波特率时钟从清零状态启动，所以这样做会导致串行比特流立即开始移出。
6. 中断控制位 UTXISELx 的设置决定何时产生发送中断。

16.3 9 位数据模式下的发送

1. 设置 UART （按**第 16.2 节 “ 在 8 位数据模式下发送 ”**所述步骤进行设置）。
2. 使能 UART。
3. 将 UTXEN 位置 1 （产生发送中断）。
4. 仅向 UxTXREG 写入 16 位的值。
5. 向 UxTXREG 写入字可触发 9 位数据向 TSR 传输。串行比特流在波特率时钟的第一个上升沿开始移出。
6. 中断控制位 UTXISELx 的设置决定何时产生发送中断。

16.4 中断和同步发送序列

下述序列会发送一个报文帧头，包括一个间隔字符和其后的一个自动波特率同步字节。

1. 将 UARTG 配置为所需的模式。
2. 将 UTXEN 和 UTXBRK 置 1 以设置间隔字符。
3. 将一个 “ 虚拟 ” 字符载入 UxTXREG 寄存器中启动发送 （值被忽略）。
4. 向 UxTXREG 写入 “ 55h ”；将同步字符装入发送 FIFO 中。
5. 发送间隔字符后，由硬件将 UTXBRK 位复位。此时发送同步字符。

16.5 在 8 位或 9 位数据模式下接收

1. 设置 UART （按**第 16.2 节 “ 在 8 位数据模式下发送 ”**所述步骤进行设置）。
2. 使能 UART。
3. 当接收到一个或多个数据字符时，将会根据中断控制位 URXISELx 的设置产生接收中断。
4. 读 OERR 位以确定是否发生了溢出错误。必须用软件将 OERR 位复位。
5. 读 UxRXREG。

读取 UxRXREG 字符的行为会将下一个字符移入接收 FIFO 的顶部，包括新的 PERR 和 FERR 值组。

16.6 $\overline{\text{UxCTS}}$ 和 $\overline{\text{UxRTS}}$ 控制引脚操作

UARTx 允许发送（$\overline{\text{UxCTS}}$）和 UARTx 请求发送（$\overline{\text{UxRTS}}$）是两个与 UART 模块有关的由硬件控制的引脚。这两个引脚允许 UART 运行在单工模式和流控制模式下。使用这两个引脚可控制 UART 模块和数据终端设备（Data Terminal Equipment， DTE）之间的数据发送和接收。可使用 UxMODE 寄存器的 UEN1:UEN0 位对这两个引脚进行配置。

16.7 红外支持

UART 模块支持两类红外 UART：一种是 IrDA 时钟输出，支持外部 IrDA 编码和解码器件 （传统模块支持），另一种是全功能的 IrDA 编码和解码器。注意，因为 IrDA 模式需要 16 倍频波特率时钟，所以它们只能工作在 BRGH 位 （UxMODE<3>）为 0 的情况下。

16.7.1 支持外部 IrDA 的 IrDA 时钟输出

要支持外部 IrDA 编码和解码器件， BCLKx 引脚 （与 $\overline{\text{UxRTS}}$ 引脚相同）可以配置用来产生 16 倍频波特率时钟。当使能了 UART 模块且 UEN1:UEN0 = 11 时，BCLKx 引脚将输出 16 倍频波特率时钟，用于支持 IrDA 编码器芯片。

16.7.2 内置 IrDA 编码器和解码器

UART 模块在其内部完全实现了 IrDA 编码器和解码器。内置 IrDA 编码器和解码器的功能可通过 IREN 位（UxMODE<12>）来使能。当使能（即 IREN = 1）时，接收引脚 （UxRX）可作为红外接收器的输入引脚。发送引脚 （UxTX）作为红外发送器的输出引脚。

寄存器 16-1： UxMODE：UARTx 模式寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
UARTEN(1)	—	USIDL	IREN(2)	RTSMD	—	UEN1	UEN0
bit 15							bit 8

R/C-0, HC	R/W-0	R/W-0, HC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7							bit 0

图注：	C = 可清零位	HC = 可由硬件清零的位	
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **UARTEN：**UARTx 使能位 (1)
1 = 使能 UARTx ； UARTx 根据 UEN1:UEN0 的定义控制所有 UARTx 引脚
0 = 禁止 UARTx ；由端口锁存器控制所有 UARTx 引脚； UARTx 的功耗最小

bit 14 **未实现：**读为 0

bit 13 **USIDL：**空闲模式下停止位
1 = 器件进入空闲模式后模块停止工作
0 = 空闲模式下模块继续工作

bit 12 **IREN：**IrDA® 编码器和解码器使能位 (2)
1 = 使能 IrDA 编码器和解码器
0 = 禁止 IrDA 编码器和解码器

bit 11 **RTSMD：**$\overline{\text{UxRTS}}$ 引脚模式选择位
1 = $\overline{\text{UxRTS}}$ 引脚处于单工模式
0 = $\overline{\text{UxRTS}}$ 引脚处于流控制模式

bit 10 **未实现：**读为 0

bit 9-8 **UEN1:UEN0：**UARTx 使能位
11 = 使能并使用 UxTX、 UxRX 和 BCLKx 引脚； $\overline{\text{UxCTS}}$ 引脚由端口锁存器控制
10 = 使能并使用 UxTX、 UxRX、 $\overline{\text{UxCTS}}$ 和 $\overline{\text{UxRTS}}$ 引脚
01 = 使能并使用 UxTX、 UxRX 和 $\overline{\text{UxRTS}}$ 引脚； $\overline{\text{UxCTS}}$ 引脚由端口锁存器控制
00 = 使能并使用 UxTX 和 UxRX 引脚； UxCTS 和 UxRTS/BCLKx 引脚由端口锁存器控制

bit 7 **WAKE：**在休眠模式下检测到起始位时使能唤醒位
1 = UARTx 将继续对 UxRX 引脚进行采样；在下降沿产生中断，在下一个上升沿由硬件将该位清零
0 = 禁止唤醒

bit 6 **LPBACK：**UARTx 环回模式选择位
1 = 使能环回模式
0 = 禁止环回模式

bit 5 **ABAUD：**自动波特率使能位
1 = 允许对下一个字符进行波特率测量——要求接收到同步字段 （55h） ；完成时由硬件清零
0 = 禁止或已完成波特率测量

注 1： 如果 UARTEN = 1，外设的输入和输出必须配置给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。
2： 此功能仅适用于 16 倍频 BRG 模式 （BRGH = 0）。

寄存器 16-1： UxMODE：UARTx 模式寄存器 （续）

bit 4	**RXINV**：接收极性反转位 1 = UxRX 的空闲状态是 0 0 = UxRX 的空闲状态是 1
bit 3	**BRGH**：高波特率使能位 1 = BRG 在每个位周期内产生 4 个时钟信号 （4 倍频波特率时钟，高速模式） 0 = BRG 在每个位周期内产生 16 个时钟信号 （16 倍频波特率时钟，标准模式）
bit 2-1	**PDSEL1:PDSEL0**：奇偶校验和数据选择位 11 = 9 位数据，无奇偶校验 10 = 8 位数据，奇校验 01 = 8 位数据，偶校验 00 = 8 位数据，无奇偶校验
bit 0	**STSEL**：停止位选择位 1 = 两个停止位 0 = 一个停止位

注 1： 如果 UARTEN = 1，外设的输入和输出必须配置给可用的 RPn 引脚。更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

2： 此功能仅适用于 16 倍频 BRG 模式 （BRGH = 0）。

寄存器 16-2： UxSTA：UARTx 状态和控制寄存器

R/W-0	R/W-0	R/W-0	U-0	R/W-0 HC	R/W-0	R-0	R-1
UTXISEL1	UTXINV(1)	UTXISEL0	—	UTXBRK	UTXEN(2)	UTXBF	TRMT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

图注：	C = 可清零位	HC = 可由硬件清零的位	
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15,13 **UTXISEL1:UTXISEL0**：发送中断模式选择位
11 = 保留；不使用
10 = 当一个字符被传输到发送移位寄存器 （TSR）导致发送缓冲器为空时，产生中断
01 = 当最后一个字符被移出发送移位寄存器；所有发送操作执行完毕时产生中断
00 = 当字符发送到发送移位寄存器时发生中断 （这暗指发送缓冲器中至少还有一个字符）时产生中断

bit 14 **UTXINV**：IrDA® 编码器发送极性反转位 (1)
IREN = 0：
1 = UxTX 空闲状态为 0
0 = UxTX 空闲状态为 1
IREN = 1：
1 = UxTX 空闲状态为 1
0 = UxTX 空闲状态为 0

bit 12 **未实现**：读为 0

bit 11 **UTXBRK**：发送间隔位
1 = 在下一次发送时发出同步间隔字符——即启动位，后跟 12 个 0，然后是停止位；完成时由硬件清零
0 = 同步间隔字符发送被禁止或发送完毕

bit 10 **UTXEN**：发送使能位 (2)
1 = 使能发送，由 UARTx 控制 UxTX 引脚
0 = 禁止发送，中止所有等待的发送，缓冲器复位。由端口控制 UxTX 引脚。

bit 9 **UTXBF**：发送缓冲器填满状态位 （只读）
1 = 发送缓冲器已满
0 = 发送缓冲器未满，至少还可写入一个或多个字符

bit 8 **TRMT**：发送移位寄存器空位 （只读）
1 = 发送移位寄存器为空，同时发送缓冲器为空 （最后一次发送已完成）
0 = 发送移位寄存器非空，发送正在进行或进入等待队列

注 1： 仅当使能了 IrDA 编码器 （IREN = 1）时，该位的值才影响模块的发送属性。
2： 如果 UARTEN = 1，外设的输入和输出必须配置给可用的 RPn 引脚。
更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

寄存器 16-2： UxSTA：UARTx 状态和控制寄存器 （续）

bit 7-6 **URXISEL1:URXISEL0**：接收中断模式选择位
11 = 在 RSR 传输使接收缓冲器满时发生中断 （即接收 4 个数据字符）
10 = 在 RSR 传输使接收缓冲器 3/4 满时发生中断 （即接收 3 个数据字符）
0x = 当接收到一个字符且 RSR 的内容被传输给接收缓冲器时，中断标志位置 1。接收缓冲器有一个或多个字符。

bit 5 **ADDEN**：地址字符检测位 （接收数据的第 8 位 = 1）
1 = 使能地址检测模式。如果没有选择 9 位模式，该位无效。
0 = 禁止地址检测模式

bit 4 **RIDLE**：接收器空闲位 （只读）
1 = 接收器空闲
0 = 接收器工作

bit 3 **PERR**：奇偶校验错误状态位 （只读）
1 = 检测到当前字符 （接收 FIFO 顶部的字符）的奇偶校验错误
0 = 没有检测到奇偶校验错误

bit 2 **FERR**：帧错误状态位 （只读）
1 = 检测到当前字符 （接收 FIFO 顶部的字符）的帧错误
0 = 没有检测到帧错误

bit 1 **OERR**：接收缓冲器溢出错误状态位 （清零 / 只读）
1 = 接收缓冲器已经溢出
0 = 接收缓冲器未溢出（将先前置 1 的 OERR 位清零（1 → 0 的跳变）会将接收缓冲器和 RSR 复位到空状态）

bit 0 **URXDA**：接收缓冲器数据可用位 （只读）
1 = 接收缓冲器中有数据，至少有一个或多个字符可被读取
0 = 接收缓冲器为空

注 1： 仅当使能了 IrDA 编码器 （IREN = 1）时，该位的值才影响模块的发送属性。
2： 如果 UARTEN = 1，外设的输入和输出必须配置给可用的 RPn 引脚。
更多信息，请参见**第 9.4 节 “ 外设引脚选择 ”**。

注：

17.0 并行主端口 (PMP)

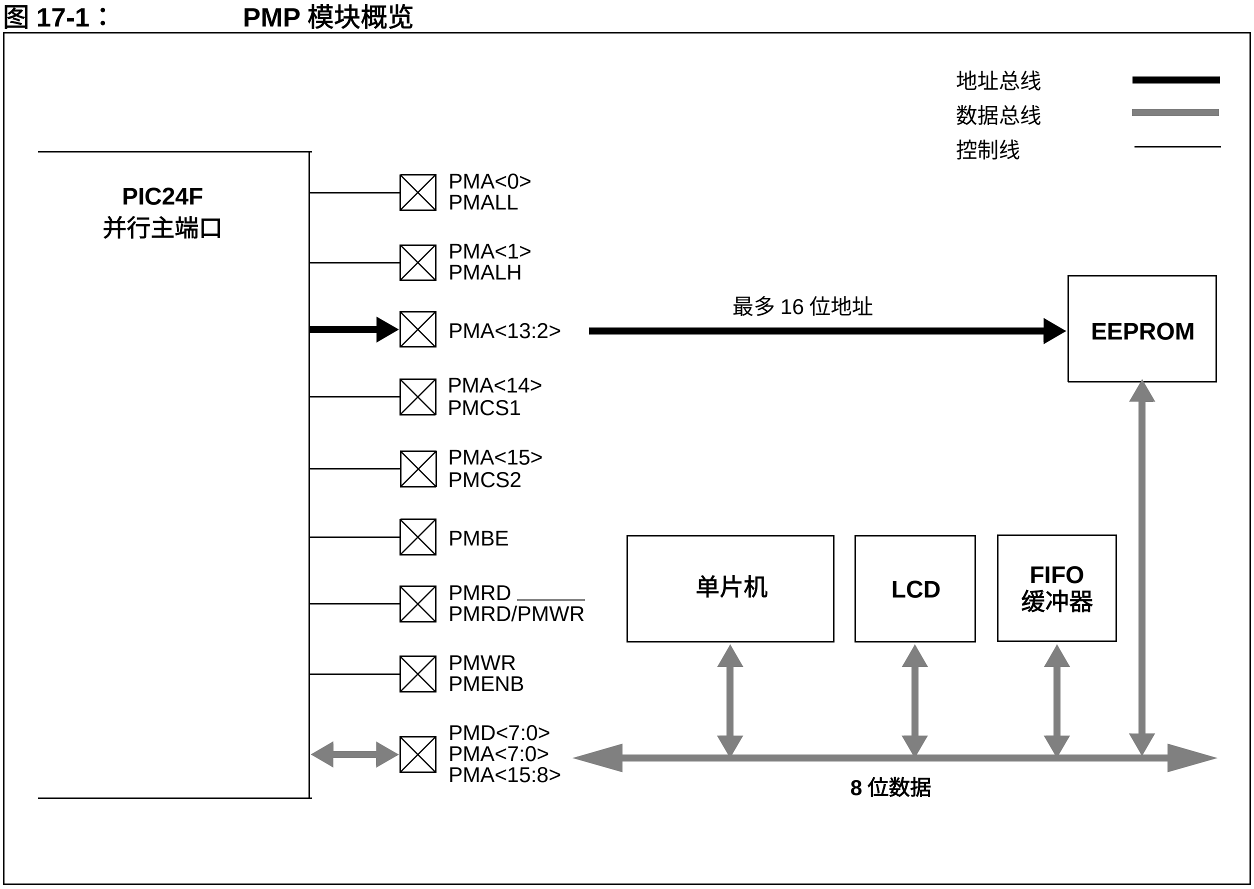
注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的"**第 13 章 并行主端口 (PMP)**"(DS39713A_CN)。

并行主端口（Parallel Master Port，PMP）模块是专为与多种并行器件，如通信外设、LCD、外部存储器和单片机等通信而设计的 8 位并行 I/O 模块。由于并行外设的接口差异很大，因此 PMP 具有很强的可配置能力。

PMP 模块的主要特性包括：

- 最多 16 条可编程地址线
- 最多 2 条片选信号线
- 可编程选通选项：
  - 独立的读和写选通或；
  - 带使能选通的读 / 写选通
- 地址自动递增 / 自动递减
- 可编程地址 / 数据复用
- 可编程控制信号的极性
- 支持传统并行从端口
- 支持增强型并行从端口：
  - 地址支持
  - 4 字节深自动递增缓冲器
- 可编程等待状态
- 可选的输入电平

图 17-1： PMP 模块概览



寄存器 17-1： PMCON：并行主端口控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0[(1)]	R/W-0[(1)]	R/W-0[(1)]	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **PMPEN**：并行主端口使能位
1 = 使能 PMP
0 = 禁止 PMP，不执行任何片外访问

bit 14 **未实现**：读为 0

bit 13 **PSIDL**：空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 模块在空闲模式下继续工作

bit 12-11 **ADRMUX1:ADRMUX0**：地址 / 数据复用选择位
11 = 保留
10 = 所有 16 位地址在 PMD<7:0> 引脚上复用
01 = 地址的低 8 位在 PMD<7:0> 引脚上复用，高 3 位在 PMA<10:8> 引脚上复用
00 = 地址和数据使用独立的引脚

bit 10 **PTBEEN**：字节使能端口使能位 （16 位主模式）
1 = 使能 PMBE 端口
0 = 禁止 PMBE 端口

bit 9 **PTWREN**：写使能选通端口使能位
1 = 使能 PMWR/PMENB 端口
0 = 禁止 PMWR/PMENB 端口

bit 8 **PTRDEN**：读 / 写选通端口使能位
1 = 使能 PMRD/PMWR 端口
0 = 禁止 PMRD/PMWR 端口

bit 7-6 **CSF1:CSF0**：片选功能位
11 = 保留
10 = PMCS1 用作片选功能
01 = 保留
00 = 保留

bit 5 **ALP**：地址锁存极性位 [(1)]
1 = 高电平有效 （PMALL 和 PMALH）
0 = 低电平有效 （PMALL 和 PMALH）

bit 4 **CS2P**：片选 2 极性位 [(1)]
1 = 高电平有效 （PMCS2/PMCS2）
0 = 低电平有效 （PMCS2/PMCS2）

bit 3 **CS1P**：片选 1 极性位 [(1)]
1 = 高电平有效 （PMCS1/PMCS1）
0 = 低电平有效 （PMCS1/PMCS1）

注 1： 这些位在相应引脚用作地址线时无效。

寄存器 17-1： PMCON：并行主端口控制寄存器 （续）

bit 2 **BEP：**字节使能极性位
1 = 字节使能高电平有效 （PMBE）
0 = 字节使能低电平有效 （$\overline{PMBE}$）

bit 1 **WRSP：**写选通极性位
对于从模式和主模式 2 （PMMODE<9:8> = 00、 01 或 10）：
1 = 写选通高电平有效 （PMWR）
0 = 写选通低电平有效 （$\overline{PMWR}$）
对于主模式 1 （PMMODE<9:8> = 11）：
1 = 使能选通高电平有效 （PMENB）
0 = 使能选通低电平有效 （$\overline{PMENB}$）

bit 0 **RDSP：**读选通极性位
对于从模式和主模式 2 （PMMODE<9:8> = 00、 01 或 10）：
1 = 读选通高电平有效 （PMRD）
0 = 读选通低电平有效 （$\overline{PMRD}$）
对于主模式 1 （PMMODE<9:8> = 11）：
1 = 读 / 写选通高电平有效 （PMRD/PMWR）
0 = 读 / 写选通低电平有效 （$\overline{PMRD/PMWR}$）

注 1： 这些位在相应引脚用作地址线时无效。

寄存器 17-2： PMMODE：并行主端口模式寄存器

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAITB1(1)	WAITB0(1)	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1(1)	WAITE0(1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **BUSY**：忙位 （仅限主模式）
- 1 = 端口忙 （当处理器暂停时无用）
- 0 = 端口不忙

bit 14-13 **IRQM1:IRQM0**：中断请求模式位
- 11 = 当读取读缓冲器3或写入写缓冲器3时产生中断（缓冲的PSP模式）或在PMA<1:0> = 11时执行读或写操作时产生中断 （仅限可寻址 PSP 模式）
- 10 = 不产生中断，处理器暂停
- 01 = 在读 / 写周期结束时产生中断
- 00 = 不产生中断

bit 12-11 **INCM1:INCM0**：递增模式位
- 11 = PSP 读和写缓冲器自动递增 （仅限传统 PSP 模式）
- 10 = 每个读 / 写周期 ADDR<10:0> 减 1
- 01 = 每个读 / 写周期 ADDR<10:0> 加 1
- 00 = 地址不发生自动增减

bit 10 **MODE16**：8/16 位模式位
- 1 = 16 位模式：数据寄存器为 16 位，对数据寄存器执行读或写操作将调用两次 8 位传输
- 0 = 8 位模式：数据寄存器为 8 位，对数据寄存器执行读或写操作将调用一次 8 位传输

bit 9-8 **MODE1:MODE0**：并行端口模式选择位
- 11 = 主模式 1 （PMCS1、 PMRD/PMWR、 PMENB、 PMBE、 PMA<x:0> 和 PMD<7:0>）
- 10 = 主模式 2 （PMCS1、 PMRD、 PMWR、 PMBE、 PMA<x:0> 和 PMD<7:0>）
- 01 = 增强型 PSP，控制信号 （PMRD、 PMWR、 PMCS1、 PMD<7:0> 和 PMA<1:0>）
- 00 = 传统并行从端口，控制信号 （PMRD、 PMWR、 PMCS1 和 PMD<7:0>）

bit 7-6 **WAITB1:WAITB0**：从数据建立到执行读 / 写的等待状态配置位 (1)
- 11 = 数据等待时间为 4 个 TCY ；地址复用时间为 4 个 TCY
- 10 = 数据等待时间为 3 个 TCY ；地址复用时间为 3 个 TCY
- 01 = 数据等待时间为 2 个 TCY ；地址复用时间为 2 个 TCY
- 00 = 数据等待时间为 1 个 TCY ；地址复用时间为 1 个 TCY

bit 5-2 **WAITM3:WAITM0**：从执行读操作到字节使能选通的等待状态配置位
- 1111 = 额外等待 15 个 TCY
- ...
- 0001 = 额外等待 1 个 TCY
- 0000 = 无额外等待周期 （强制操作在 1 个 TCY 内完成） (2)

bit 1-0 **WAITE1:WAITE0**：选通后数据保持的等待状态配置位 (1)
- 11 = 等待 4 个 TCY
- 10 = 等待 3 个 TCY
- 01 = 等待 2 个 TCY
- 00 = 等待 1 个 TCY

注 1： 只要 WAITM3:WAITM0 = 0000， WAITB 和 WAITE 位就可被忽略。
2： 连续读和 / 或写操作之间需要一个周期的延时。

寄存器 17-3： PMADDR：并行主端口地址寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CS2	CS1	ADDR<13:8>					
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADDR<7:0>							
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **CS2**：片选 2 位
1 = 片选 2 有效
0 = 片选 2 无效

bit 14 **CS1**：片选 1 位
1 = 片选 1 有效
0 = 片选 1 无效

bit 13-0 **ADDR13:ADDR0**：并行端口目标地址位

寄存器 17-4： PMAEN：并行主端口使能寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **PTEN15:PTEN14**：PMCSx 选通使能位
1 = PMA15 和 PMA14 用作 PMA<15:14> 或分别用作 PMCS2 和 PMCS1
0 = PMA15 和 PMA14 用作端口 I/O

bit 13-2 **PTEN13:PTEN2**：PMP 地址端口使能位
1 = PMA<13:2> 作为 PMP 地址线
0 = PMA<13:2> 作为端口 I/O

bit 1-0 **PTEN1:PTEN0**：PMALH/PMALL 选通使能位
1 = PMA1 和 PMA0 用作 PMA<1:0> 或分别用作 PMALH 和 PMALL
0 = PMA1 和 PMA0 引脚用作端口 I/O

寄存器 17-5： PMSTAT：并行主端口状态寄存器

R-0	R/W-0, HS	U-0	U-0	R-0	R-0	R-0	R-0
IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F
bit 15							bit 8

R-1	R/W-0, HSS	U-0	U-0	R-1	R-1	R-1	R-1
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7							bit 0

图注：	HS = 可由硬件置 1 的位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **IBF：**输入缓冲器满状态位
1 = 所有可写的输入缓冲寄存器为满
0 = 部分或所有可写的输入缓冲器寄存器为空

bit1 4 **IBOV：**输入缓冲器溢出状态位
1 = 尝试对已满的输入字节寄存器执行写操作 （必须由软件清零）
0 = 未发生溢出

bit 13-12 **未实现：**读为 0

bit 11-8 **IB3F:IB0F** 输入缓冲器 x 满状态位
1 = 输入缓冲器包含尚未读取的数据 （读缓冲器将清零此位）
0 = 输入缓冲器不包含任何未读取的数据

bit 7 **OBE：**输入缓冲器空状态位
1 = 所有可读的输出缓冲寄存器均为空
0 = 部分或所有可读的输出缓冲寄存器均满

bit 6 **OBUF：**输出缓冲器下溢状态位
1 = 对空输出字节寄存器执行读操作 （必须由软件清零）
0 = 未发生下溢

bit 5-4 **未实现：**读为 0

bit 3-0 **OB3E:OB0E：**输出缓冲器 x 空状态位
1 = 输出缓冲器为空 （将数据写入该缓冲器会将该位清零）
0 = 输出缓冲器中包含未被发送的数据

寄存器 17-6： PADCFG1：焊垫配置控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	RTSEC-SEL[(1)]	PMPTTL
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-2 **未实现：**读为 0

bit 1 **RTSECSEL：**RTCC 秒时钟输出选择位 [(1)]
1 = 选择 RTCC 引脚输出 RTCC 秒时钟
0 = 选择 RTCC 引脚输出 RTCC 闹钟脉冲

bit 0 **PMPTTL：**PMP 模块 TTL 输入缓冲器选择位
1 = PMP 模块输入 （PMDx 和 PMCS1）使用 TTL 输入缓冲器
0 = PMP 模块输入使用施密特触发器输入缓冲器

注 1： 要使能实际 RTCC 输出，需将 RTCOE （RCFGCAL<10>）位置 1。

图 17-2： 传统并行从端口示例

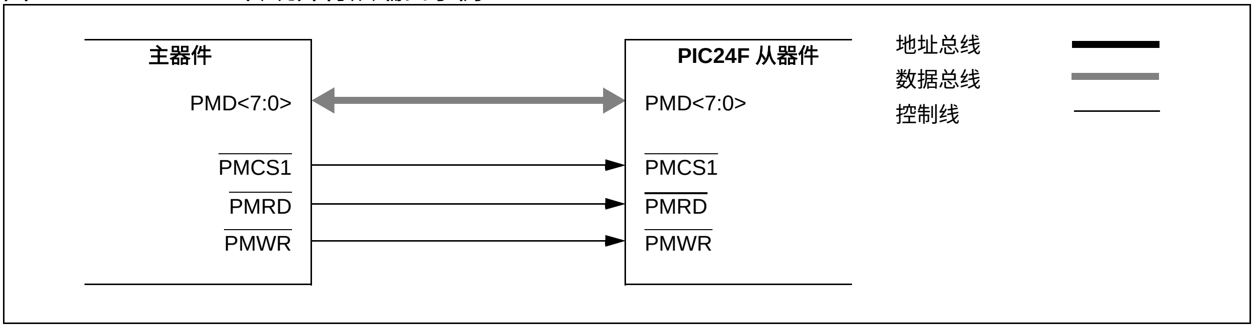


图 17-3： 可寻址的并行从端口示例

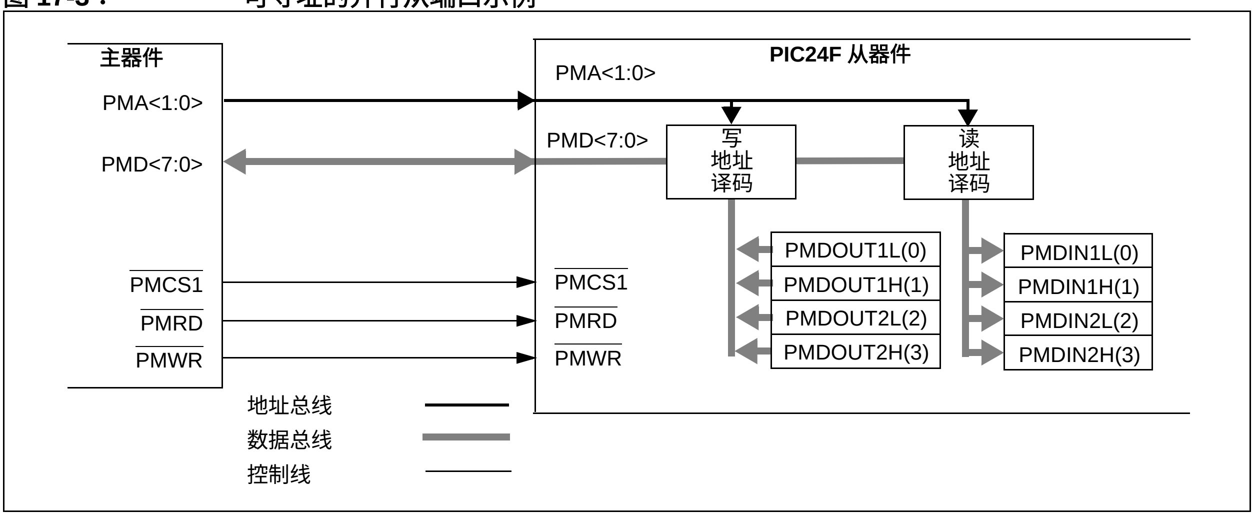


表 17-1： 从模式地址解析方案

PMA<1:0>	输出寄存器 （缓冲器）	输入寄存器 （缓冲器）
00	PMDOUT1<7:0> (0)	PMDIN1<7:0> (0)
01	PMDOUT1<15:8> (1)	PMDIN1<15:8> (1)
10	PMDOUT2<7:0> (2)	PMDIN2<7:0> (2)
11	PMDOUT2<15:8> (3)	PMDIN2<15:8> (3)

图 17-4： 主模式、未复用的寻址 （独立的读和写选通、两个片选信号）

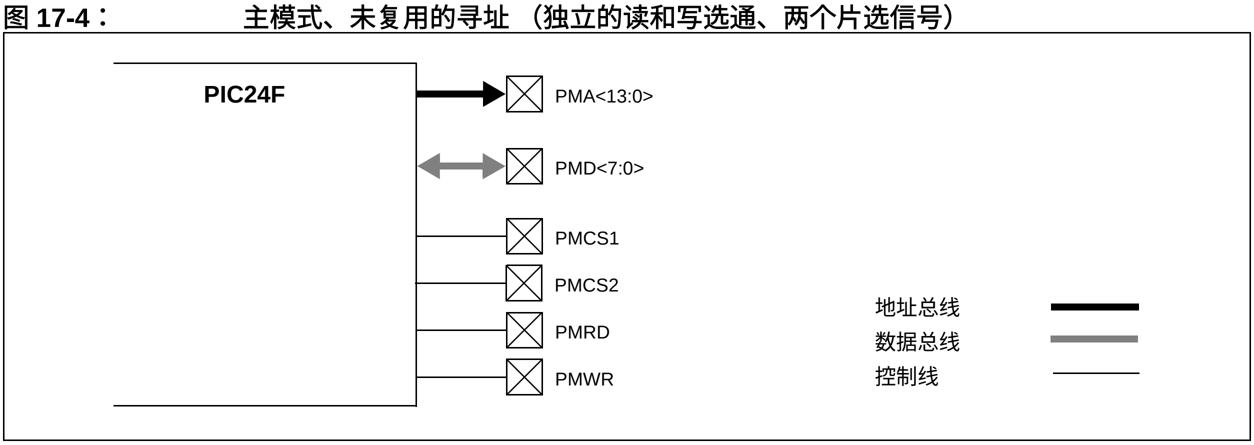


图 17-5： 主模式、部分复用寻址 （独立的读和写选通、两个片选信号）

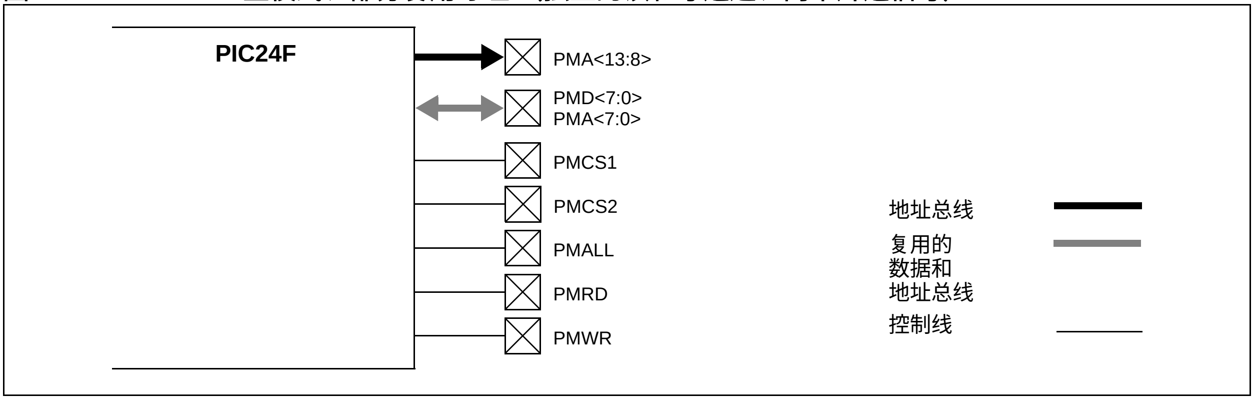


图 17-6： 主模式、完全复用寻址 （独立的读和写选通、两个片选信号）

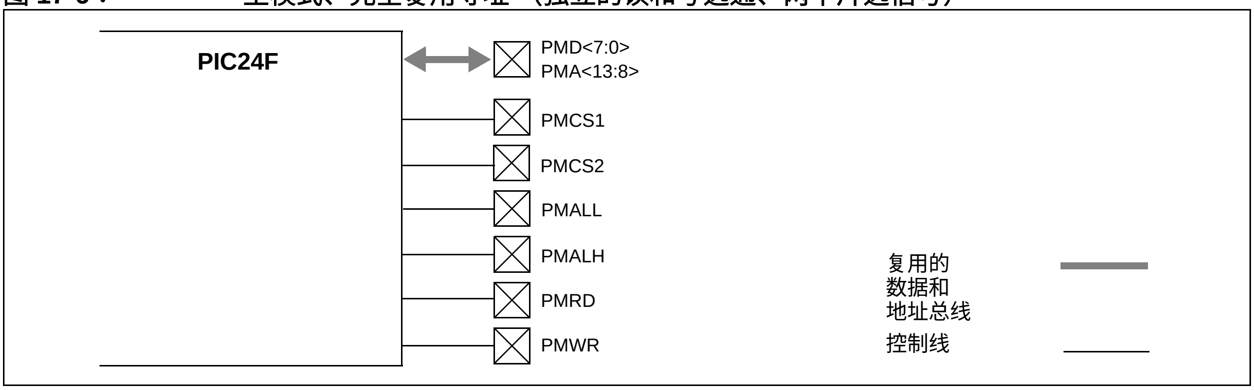


图 17-7： 复用寻址应用示例

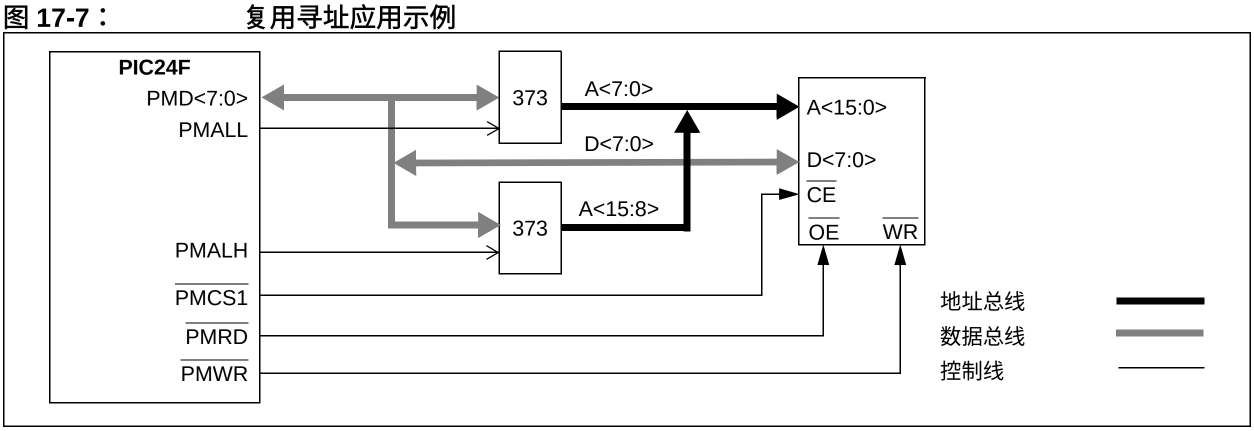


图 17-8： 部分复用寻址应用示例

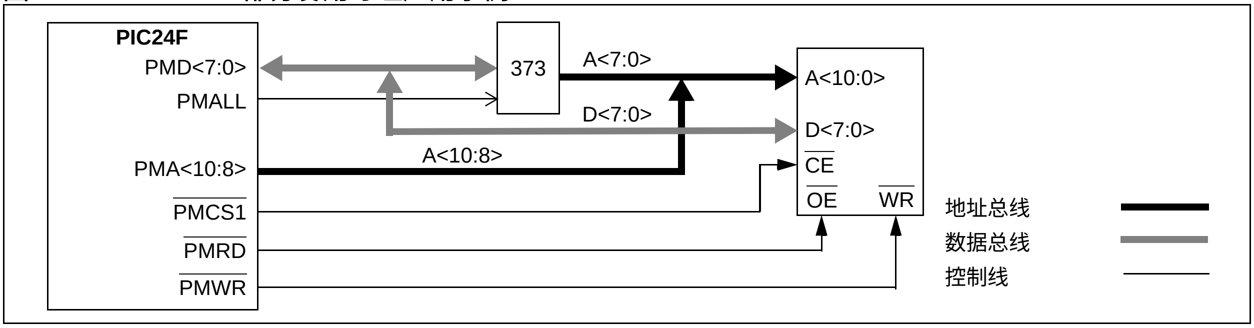


图 17-9： 8 位地址和数据复用的应用示例

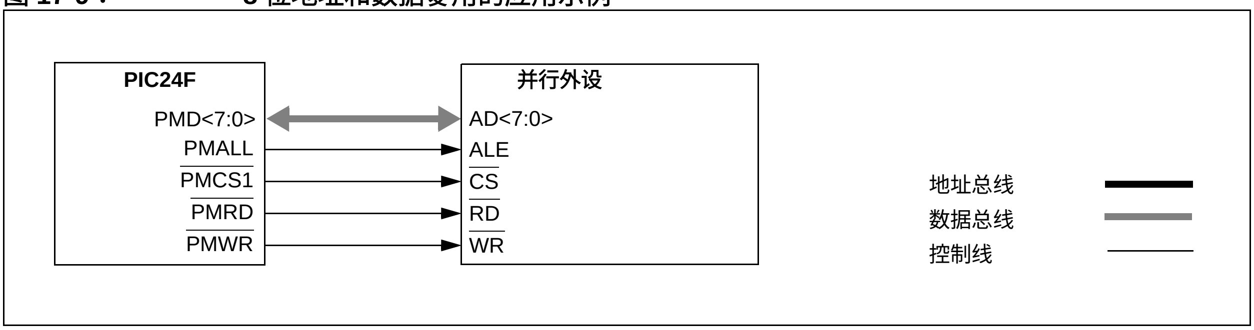


图 17-10： 并行 EEPROM 示例 （最多 15 位地址 8 位数据）

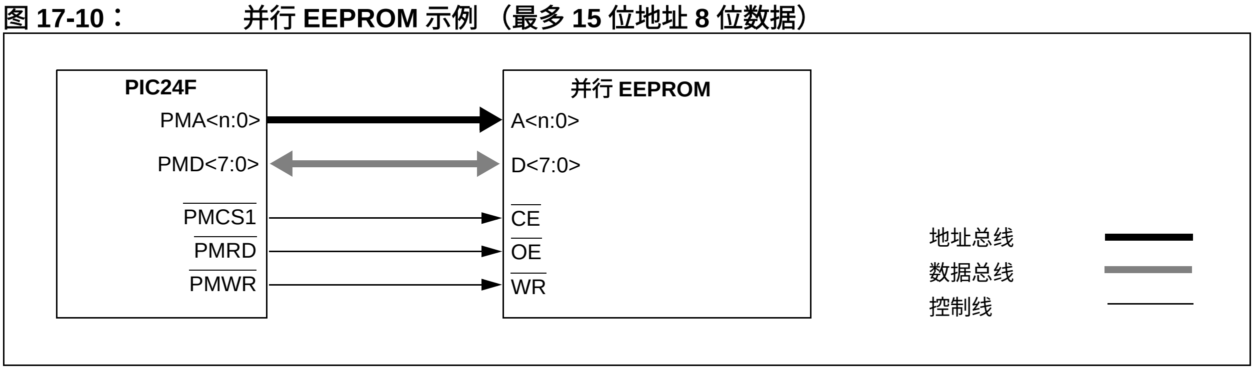


图 17-11： 并行 EEPROM 示例 （最多 15 位地址 16 位数据）

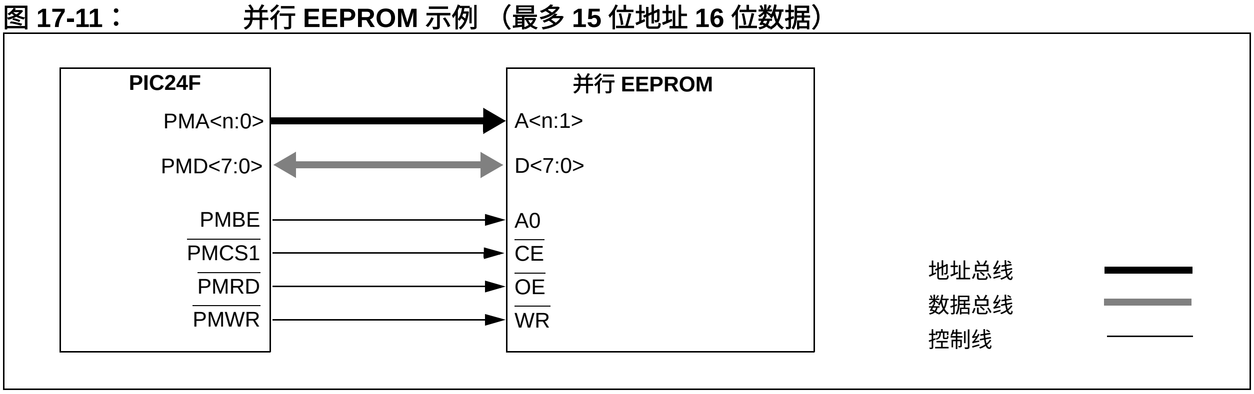
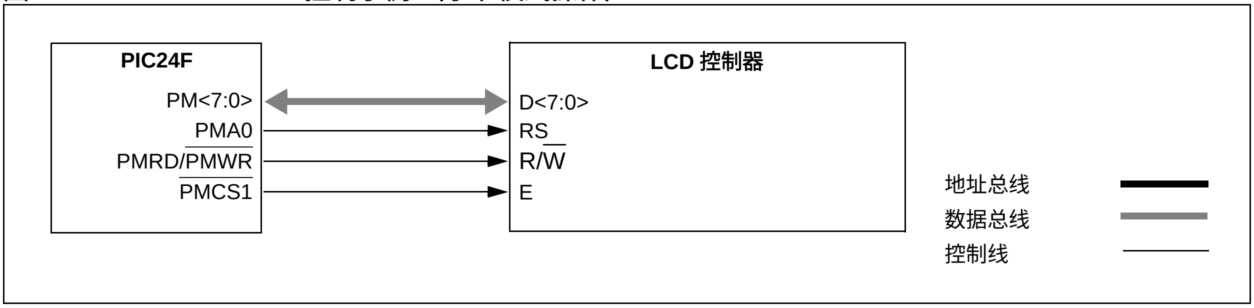


图 17-12： LCD 控制示例 （字节模式操作）

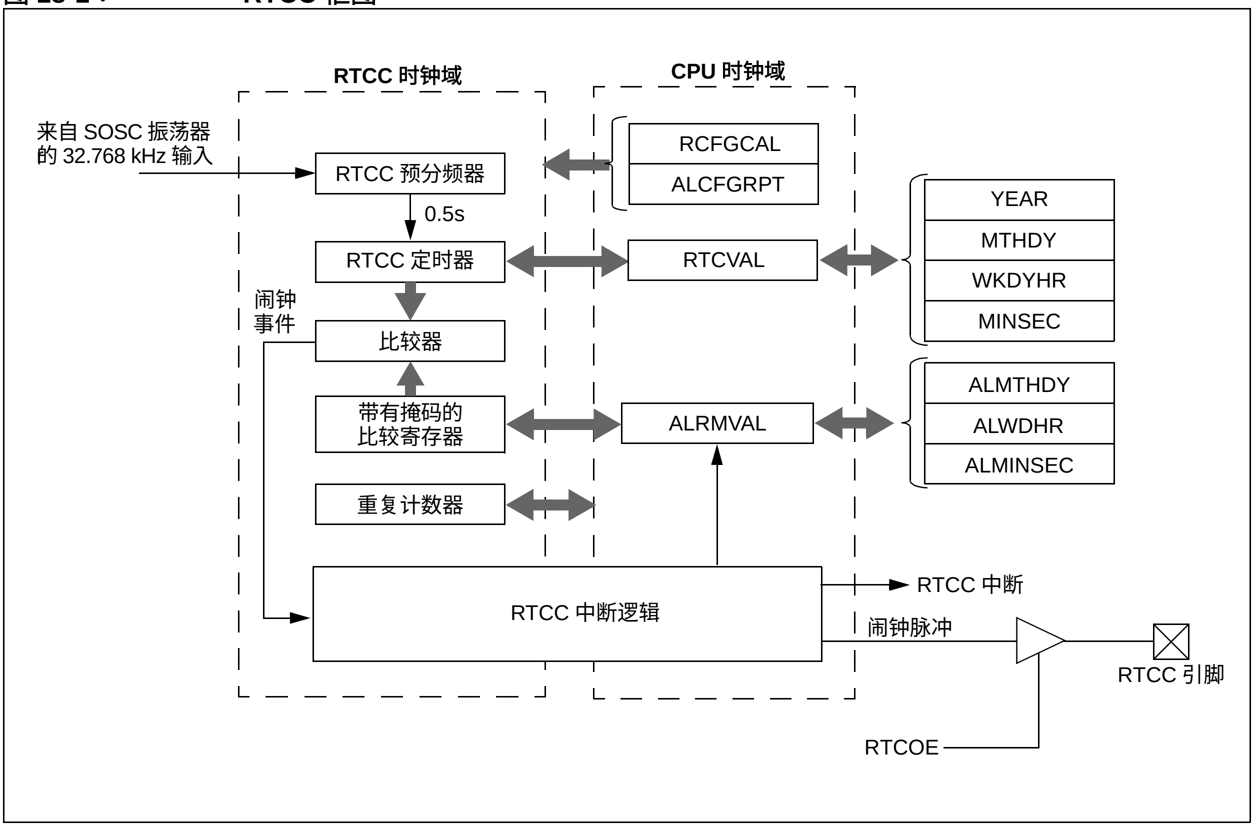


注：

18.0 实时时钟和日历 （RTCC）

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的“**第 29 章 实时时钟和日历（RTCC）**”（DS39696A_CN）。

图 18-1： RTCC 框图



18.1 RTCC 模块寄存器

RTCC 模块寄存器分为三类：

- RTCC 控制寄存器
- RTCC 值寄存器
- 闹钟值寄存器

18.1.1 寄存器映射

要限制寄存器接口，可通过相应的寄存器指针访问RTCC定时器和闹钟寄存器。RTCC 值寄存器窗口（RTCVALH 和 RTCVALL）使用 RTCPTR 位（RCFGCAL<9:8>）选择所需定时器寄存器对 （见表 18-1）。

通过写入RTCVALH字节，RTCC指针值RTCPTR<1:0>位递减直到 00，达到 00 后在手动更改指针值前，可通过 RTCVALH 和 RTCVALL 来访问 MINUTES 和 SECONDS 的值。

表 18-1： RTCVAL 寄存器映射

RTCPTR <1:0>	RTCC 值寄存器窗口	
	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

闹钟值寄存器窗口（ALRMVALH 和 ALRMVALL）使用 ALRMPTR 位 （ALCFGRPT<9:8>）来选择所需闹钟寄存器对 （见表 18-2）。

通过写入 ALRMVALH 字节，闹钟指针值（ALRMPTR<1:0>）位递减直到 00，达到 00 后在手动更改指针值前，可通过 ALRMVALH 和 ALRMVALL 来访问 ALRMMIN 和 ALRMSEC 的值。

表 18-2： ALRMVAL 寄存器映射

ALRMPTR <1:0>	闹钟值寄存器窗口	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

考虑到 16 位内核不能区分 8 位和 16 位读操作，用户必须注意读 ALRMVALH 和 ALRMVALL 字节会递减 ALRMPTR<1:0> 的值。这同样适用于 RTCVALH 或 RTCVALL 字节，读这两个字节 RTCPTR<1:0> 会递减。

> **注：** 这仅适用于读操作，不适用于写操作。

18.1.2 写锁定

为了对任何 RTCC 定时器寄存器执行写操作，必须将 RTCWREN 位（RCFGCAL<13>）置 1（见例 18-1）。

> **注：** 为避免意外写入定时器，建议在除此之外的任何时候都保持 RTCWREN 位（RCFGCAL<13>）清零。对于要使 RTCWREN 位置 1 的操作，由于在解锁序列和 RTCWREN 置 1 之间仅允许一个指令周期的时间段；因此，建议按照 例 18-1 中的过程执行代码。
> 对于用 C 语言编写的应用程序，应使用行内汇编执行解锁序列。

例 18-1： 将 RTCWREN 位置 1

```
asm volatile("disi #5");
asm volatile("mov #0x55, w7");
asm volatile("mov w7, _NVMKEY");
asm volatile("mov #0xAA, w8");
asm volatile("mov w8, _NVMKEY");
asm volatile("bset _RCFGCAL, #13");      //set the RTCWREN bit
```


18.1.3 RTCC 控制寄存器

寄存器 18-1： RCFGCAL：RTCC 校准和配置寄存器 (1)

R/W-0	U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
RTCEN(2)	—	RTCWREN	RTCSYNC	HALFSEC(3)	RTCOE	RTCPTR1	RTCPTR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **RTCEN**：RTCC 使能位 (2)
1 = 使能 RTCC 模块
0 = 禁止 RTCC 模块

bit 14 **未实现**：读为 0

bit 13 **RTCWREN**：RTCC 值寄存器写使能位
1 = 用户可写入 RTCVALH 和 RTCVALL 寄存器
0 = RTCVALH 和 RTCVALL 寄存器被锁定以阻止用户写入

bit 12 **RTCSYNC**：RTCC 值寄存器读同步位
1 = 由于计满返回，RTCVALH、RTCVALL 和 ALCFGRPT 寄存器在读操作过程中可能改变，从而导致了无效的数据读取。如果读取两次寄存器得到的数据相同，则假设数据有效。
0 = RTCVALH、 RTCVALL 或 ALCFGRPT 寄存器在读取时无需考虑计满返回

bit 11 **HALFSEC**：半秒状态位 (3)
1 = 一秒的后半秒
0 = 一秒的前半秒

bit 10 **RTCOE**：RTCC 输出使能位
1 = 使能 RTCC 输出
0 = 禁止 RTCC 输出

bit 9-8 **RTCPTR1:RTCPTR0**：RTCC 值寄存器窗口指针位
当读 RTCVALH 和 RTCVALL 寄存器时，指向相应的 RTCC 值寄存器；每次读或写 RTCVALH 时 RTCPTR<1:0> 的值都会递减，直到其到达 00。

RTCVAL<15:8>：
00 = MINUTES
01 = WEEKDAY
10 = MONTH
11 = 保留

RTCVAL<7:0>：
00 = SECONDS
01 = HOURS
10 = DAY
11 = YEAR

注 1： RCFGCAL 寄存器仅受到 POR 影响。
2： 只有 RTCWREN = 1 时才允许写入 RTCEN 位。
3： 该位是只读的。在写入 MINSEC 寄存器低半部分时，它被清零。

寄存器 18-1： RCFGCAL：RTCC 校准和配置寄存器 (1) （续）

bit 7-0 **CAL7:CAL0：**RTC 漂移校准位
`01111111` = 最大正调整，每分钟增加 508 个 RTC 时钟脉冲
...
`01111111` = 最小正调整，每分钟增加 4 个 RTC 时钟脉冲
`00000000` = 不调整
`11111111` = 最小负调整，每分钟减少 4 个 RTC 时钟脉冲
...
`10000000` = 最大负调整，每分钟减少 512 个 RTC 时钟脉冲

注 1： RCFGCAL 寄存器仅受到 POR 影响。
2： 只有 RTCWREN = 1 时才允许写入 RTCEN 位。
3： 该位是只读的。在写入 MINSEC 寄存器低半部分时，它被清零。

寄存器 18-2： PADCFG1：焊垫配置控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	RTSECSEL(1)	PMPTTL
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-2 **未实现：**读为 `0`

bit 1 **RTSECSEL：**RTCC 秒时钟输出选择位 (1)
`1` = 选择 RTCC 引脚输出 RTCC 秒时钟
`0` = 选择 RTCC 引脚输出 RTCC 闹钟脉冲

bit 0 **PMPTTL：**PMP 模块 TTL 输入缓冲器选择位
`1` = PMP 模块输入 （PMDx 和 PMCS1）使用 TTL 输入缓冲器
`0` = PMP 模块输入使用施密特触发器输入缓冲器

注 1： 要使能实际 RTCC 输出，需将 RTCOE （RCFGCAL<10>）位置 1。

寄存器 18-3： ALCFGRPT：闹钟配置寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **ALRMEN：**闹钟使能位
1 = 使能闹钟 （只要 ARPT<7:0> = 00h 且 CHIME = 0 就会在闹钟事件后自动清零）
0 = 闹钟禁止

bit 14 **CHIME：**响铃使能位
1 = 使能响铃；允许 ARPT<7:0> 位从 00h 进位到 FFh
0 = 禁止响铃； ARPT<7:0> 达到 00h 时停止

bit 13-10 **AMASK3:AMASK0：**报警掩码配置位
0001 = 每半秒
0001 = 每秒
0010 = 每 10 秒
0011 = 每分钟
0100 = 每 10 分钟
0101 = 每小时
0110 = 每天一次
0111 = 每周一次
1000 = 每月一次
1001 = 每年一次 （配置为 2 月 29 日除外，这种情况下每四年一次）
101x = 保留—不要使用
11xx = 保留—不要使用

bit 9-8 **ALRMPTR1:ALRMPTR0：**闹钟值寄存器窗口指针位
当读 ALRMVALH 和 ALRMVALL 寄存器时，指向相应的闹钟值寄存器；每次读或写 ALRMVALH 时 ALRMPTR<1:0> 的值都会递减，直到其到达 00。
<u>ALRMVAL<15:8>：</u>
00 = ALRMMIN
01 = ALRMWD
10 = ALRMMNTH
11 = 未实现
<u>ALRMVAL<7:0>：</u>
00 = ALRMSEC
01 = ALRMHR
10 = ALRMDAY
11 = 未实现

bit 7-0 **ARPT7:ARPT0：**闹钟重复计数器值位
11111111 = 闹钟将再重复 255 次
...
00000000 = 闹钟不重复
计数器在闹钟事件后递减 1。只有在 CHIME = 1 的情况下，计数器才能从 00h 进位到 FFh。

18.1.4 RTCVAL 寄存器映射

寄存器 18-4： YEAR：年值寄存器 (1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-8 **未实现：**读为 0
bit 7-4 **YRTEN3:YRTEN0：**年的十位数字的 BCD 码；包含一个从 0 到 9 的值
bit 3-0 **YRONE3:YRONE0：**年的个位数字的 BCD 码；包含一个从 0 到 9 的值

注 1： 只有 RTCWREN = 1 时才允许写入 YEAR 寄存器。

寄存器 18-5： MTHDY：月和日值寄存器 (1)

U-0	U-0	U-0	R-x	R-x	R-x	R-x	R-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-13 **未实现：**读为 0
bit 12 **MTHTEN0：**月的十位数字的 BCD 码；包含为 0 或 1 的值
bit 11-8 **MTHONE3:MTHONE0：**月的个位数字的 BCD 码；包含一个从 0 到 9 的值
bit 7-6 **未实现：**读为 0
bit 5-4 **DAYTEN1:DAYTEN0：**天的十位数字的 BCD 码；包含一个从 0 到 3 的值
bit 3-0 **DAYONE3:DAYONE0：**天的个位数字的 BCD 码；包含一个从 0 到 9 的值

注 1： 只有 RTCWREN = 1 时才允许写入该寄存器。

寄存器 18-6： WKDYHR：星期和小时值寄存器 (1)

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-11	**未实现：**读为 0
bit 10-8	**WDAY2:WDAY0：**星期数字的 BCD 码；包含一个从 0 到 6 的值
bit 7-6	**未实现：**读为 0
bit 5-4	**HRTEN1:HRTEN0：**小时的十位数字的 BCD 码；包含一个从 0 到 2 的值
bit 3-0	**HRONE3:HRONE0：**小时的个位数字的 BCD 码；包含一个从 0 到 9 的值

注 1： 只有 RTCWREN = 1 时才允许写入该寄存器。

寄存器 18-7： MINSEC：分钟和秒值寄存器

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15	**未实现：**读为 0
bit 14-12	**MINTEN2:MINTEN0：**分钟的十位数字的 BCD 码；包含一个从 0 到 5 的值
bit 11-8	**MINONE3:MINONE0：**分钟的个位数字的 BCD 码；包含一个从 0 到 9 的值
bit 7	**未实现：**读为 0
bit 6-4	**SECTEN2:SECTEN0：**秒的十位数字的 BCD 码；包含一个从 0 到 5 的值
bit 3-0	**SECONE3:SECONE0：**秒的个位数字的 BCD 码；包含一个从 0 到 9 的值

18.1.5 ALRMVAL 寄存器映射

寄存器 18-8： ALMTHDY：闹钟月和日值寄存器 (1)

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-13 **未实现：**读为 0
bit 12 **MTHTEN0：**月的十位数字的 BCD 码；包含为 0 或 1 的值
bit 11-8 **MTHONE3:MTHONE0：**月的个位数字的 BCD 码；包含一个从 0 到 9 的值
bit 7-6 **未实现：**读为 0
bit 5-4 **DAYTEN1:DAYTEN0：**天的十位数字的 BCD 码；包含一个从 0 到 3 的值
bit 3-0 **DAYONE3:DAYONE0：**天的个位数字的 BCD 码；包含一个从 0 到 9 的值

注 1： 只有 RTCWREN = 1 时才允许写入该寄存器。

寄存器 18-9： ALWDHR：闹钟星期和小时值寄存器 (1)

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-11 **未实现：**读为 0
bit 10-8 **WDAY2:WDAY0：**星期数字的 BCD 码；包含一个从 0 到 6 的值
bit 7-6 **未实现：**读为 0
bit 5-4 **HRTEN1:HRTEN0：**小时的十位数字的 BCD 码；包含一个从 0 到 2 的值
bit 3-0 **HRONE3:HRONE0：**小时的个位数字的 BCD 码；包含一个从 0 到 9 的值

注 1： 只有 RTCWREN = 1 时才允许写入该寄存器。

寄存器 18-10： ALMINSEC：闹钟分钟和秒值寄存器

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15	**未实现：**读为 0
bit 14-12	**MINTEN2:MINTEN0：**分钟的十位数字的 BCD 码；包含一个从 0 到 5 的值
bit 11-8	**MINONE3:MINONE0：**分钟的个位数字的 BCD 码；包含一个从 0 到 9 的值
bit 7	**未实现：**读为 0
bit 6-4	**SECTEN2:SECTEN0：**秒的十位数字的 BCD 码；包含一个从 0 到 5 的值
bit 3-0	**SECONE3:SECONE0：**秒的个位数字的 BCD 码；包含一个从 0 到 9 的值

18.2 校准

实时晶振输入可使用周期性自动调整功能来校准。正确校准后，RTCC 可实现每月小于 3 秒的误差。这可通过查找每分钟误差时钟脉冲数并将值存储到RCFGCAL寄存器的低半部分完成。将装入 RCFGCAL 低位部分的 8 位符号值乘以 4，每分钟一次从 RTCC 定时器中加上或减去。请参见以下步骤校准 RTCC：

1. 使用器件上的另一个定时器资源，用户必须查找到 32.768 kHz 晶振的误差。
2. 一旦找到错误，就必须转成每分钟误差时钟脉冲数并载入 RCFGCAL 寄存器。

公式 18-1： RTCC 校准

(理想频率 † – 测量频率)* 60 ＝每分钟误差时钟数
† 理想频率 = 32,768 Hz

3. a) 如果振荡器快于理想值 （第二步中的值为负），那么 RCFGCAL 寄存器值需为负。这可导致每分钟从定时器计数器中减去指定的时钟脉冲数。

   b) 如果振荡器慢于理想值 （第二步中的值为正），那么 RCFGCAL 寄存器值需为正。这可导致每分钟在定时器计数器上加上指定的时钟脉冲数。
4. 将每分钟误差时钟数除以4可得到正确的CAL值并将正确值装入 RCFGCAL 寄存器。

   （CAL 的值每递增一位就会添加或减去 4 个脉冲）。

仅当定时器关闭或紧接秒脉冲的上升沿时，才可写入到 RCFGCAL 寄存器的低位部分。

注：	用户自行决定误差值是否包含晶振初始误差、由温度引起的漂移以及晶体老化引起的漂移。

18.3 闹钟

- 可配置为从半秒到一年
- 使用 ALRMEN 位 （ALCFGRPT<15>，寄存器 18-3）使能
- 可用一次性闹钟和重复闹钟选项

18.3.1 配置闹钟

使用 ALRMEN 位使能闹钟功能。闹钟响起后该位清零。只有在 ALRMEN = 0 时才允许写入到 ALRMVAL。

如图 18-2 所示，可通过 AMASK 位（ALCFGRPT<13:10>）配置闹钟间隔。这些位确定哪些以及多少闹钟位必须与时钟值匹配，闹钟才能够发生。

闹钟也可配置为以预配置间隔重复。使能闹钟后，闹钟响起的次数存储在 ARPT 位，ARPT7:ARPT0（ALCFGRPT<7:0>）。当 ARPT 位的值等于 00h 且 CHIME 位（ALCFGRPT<14>）清零时，禁止重复功能，且只发生一次闹钟。通过将 FFh 装入 ARPT7:ARPT0，闹钟最多可重复 255 次。

每发生一次闹钟， ARPT 位的值就减 1。一旦其值到达 00h，就会发出最后一次闹钟，之后 ALRMEN 会自动清零，然后关闭闹钟。

如果 CHIME 位 = 1，可无限次重复闹钟。若 CHIME 置 1，当 ARPT 位的值达到 00 时，不禁止闹钟，而是进位到 FF 并继续无限次地计数。

18.3.2 闹钟中断

每一次闹钟事件发生时都会产生中断。此外会提供闹钟脉冲输出，其频率是闹钟频率的一半。该输出与 RTCC 时钟完全同步，且可用作其他外设的触发时钟。

> **注：** 当闹钟使能 （ALRMEN = 1）时，更改除 RCFGCAL 和 ALCFGRPT 寄存器以外的任何寄存器以及 CHIME 位都将导致误闹钟事件，从而引起错误闹钟中断。为避免误闹钟事件，应仅在禁止闹钟 （ALRMEN = 0）时才更改定时器和闹钟值。建议在 RTCSYNC = 0 时更改 ALCFGRPT 寄存器和 CHIME 位。

图 18-2： 闹钟掩码设置

闹钟掩码设置（AMASK3:AMASK0）	星期	月		日		小时		分钟		秒	
0000——每半秒 0001——每秒											
0010——每 10 秒											s
0011——每分钟										s	s
0100——每 10 分钟									m	s	s
0101——每小时								m	m	s	s
0110——每天						h	h	m	m	s	s
0111——每周	d					h	h	m	m	s	s
1000——每月				d	d	h	h	m	m	s	s
1001——每年 (1)		m	m	d	d	h	h	m	m	s	s

注 1： 每年，除了配置为 2 月 29 日外。

19.0 可编程循环冗余校验 （CRC）发生器

> **注：** 本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的 " **第 30 章 可编程循环冗余校验 （CRC）** " （DS39714A_CN）。

可编程循环冗余校验 （Cyclic Redundancy Check，CRC）发生器具有以下特性：

- 用户可编程 CRC 多项式方程
- 中断输出
- 数据 FIFO

模块实现了一个可由软件配置的 CRC 发生器。可分别使用 X15:X1 （CRCXOR<15:1>）位和 PLEN3:PLEN0 （CRCCON<3:0>）位来设置多项式的项和长度。

考虑 CRC 方程：

$$x^{16} + x^{12} + x^5 + 1$$

在 CRC 发生器中编程该多项式，应按照表 19-1 所示来设置 CRC 寄存器的位。

表 19-1： CRC 设置示例

位名	位值
PLEN3:PLEN0	1111
X15:X1	000100000010000

请注意根据该方程的要求，应将 X<15:1> 值的第 12 位和第 5 位置 1。方程的第 0 位要求是其余位的异或值。对于 16 次多项式，第 16 位也被认为是其余位的异或值，因此 X<15:1> 位不包含第 0 位或第 16 位。

图 19-2 给出了标准 CRC 发生器的结构。

图 19-1： CRC 移位器详细结构

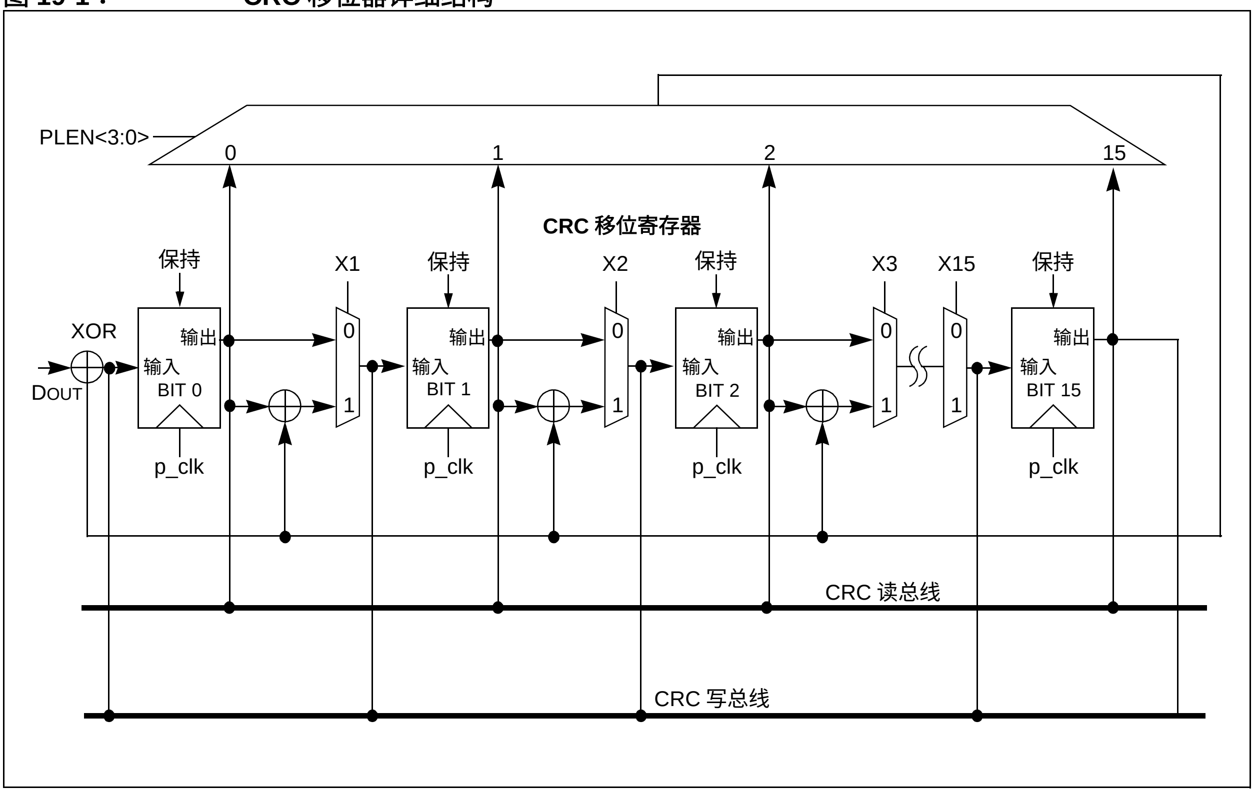
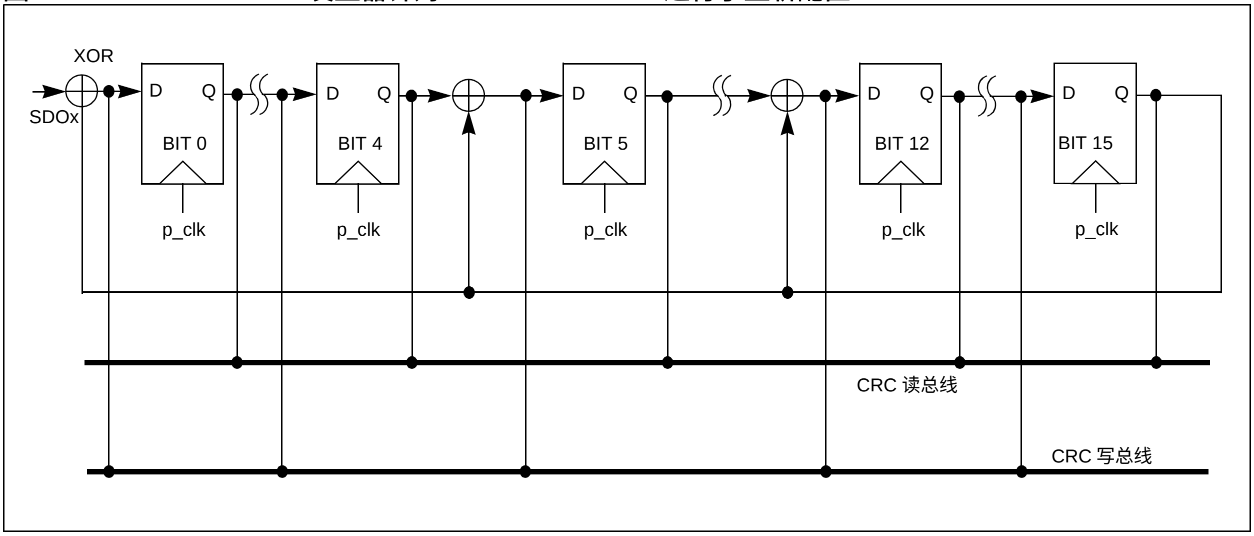


图 19-2： CRC 发生器针对 $x^{16} + x^{12} + x^5 + 1$ 进行了重新配置



19.1 用户接口

19.1.1 数据接口

要启动串行移位，必须将 CRCGO 位置 1。

该模块配备有一个 FIFO 缓冲器，当 PLEN （CRCCON<3:0>） > 7 时，FIFO 为 8 级深，其他情况下为 16 级深。首先将要计算 CRC 的数据写入 FIFO。可被写入 FIFO 的最小数据长度为一个字节。例如，如果 PLEN = 5，那么数据的长度为 PLEN + 1 = 6。该数据必须以如下方式写入：

data[5:0] = crc_input[5:0]

data[7:6] = ‘bxx

数据一旦被写入 CRCWDAT 的最高位 （由 PLEN 定义），VWORD （CRCCON<12:8>）的值将递增一。当 CRCGO = 1且VWORD > 0时，串行移位器开始向CRC 引擎移动数据。当最高位移出后，VWORD 递减一。串行移位器会继续移出数据直到 VWORD 达到 0。因此对于给定的 PLEN 值，需要 (PLEN + 1) * VWORD 个时钟周期才能完成 CRC 计算。

当 VWORD 达到 8 （或 16）时， CRCFUL 位将置 1。当 VWORD 达到 0 时， CRCMPT 位将置 1。

要将数据持续地送入 CRC 引擎，建议的操作模式是先以足够数量的数据字 “ 预先 ” 将 FIFO 填满，这样在写下一个字之前将不会产生中断。完成后，通过将 CRCGO 位置 1 启动 CRC。从此时起，应查询 VWORD 位。如果读取的字节数小于 8 或 16，则可向 FIFO 写入另一个字。

要将写入 FIFO 的字清空，必须将 CRCGO 位置 1，并且允许 CRC 移位器运行直到 CRCMPT 位置 1。

同样，为了读取正确的 CRC 值，在读取 CRCWDAT 寄存器前必须等待 CRCMPT 位变成高电平。

如果当 CRCFUL 位置 1 时写入了一个字，则 VWORD 指针将返回至 0。随后硬件将如同 FIFO 为空那样进行操作。但是，由于并未满足产生中断的条件；因此不会产生中断 （见第 **19.1.2 节 “ 中断操作 ”**）。

在写入 CRCWDAT 之后，必须至少经过一个指令周期才可以读 VWORD 位。

19.1.2 中断操作

当VWORD4:VWORD0位的值从1变为0时，产生中断。

19.2 节能模式下的操作

19.2.1 休眠模式

如果器件在模块运行过程中进入休眠模式，模块将暂停工作，保持当前状态直到时钟恢复工作。

19.2.2 空闲模式

要在空闲模式下继续完整的模块操作，必须在进入空闲模式前将 CSIDL 位清零。

如果 CSIDL = 1，则模块的表现和在休眠模式下一样；但是即使模块时钟不可用，待处理的中断事件也会被传递给 CPU。

 超前信息

19.3 寄存器

有四个寄存器用于控制可编程 CRC 的操作：

- CRCCON
- CRCXOR
- CRCDAT
- CRCWDAT

寄存器 19-1： CRCCON：CRC 控制寄存器

U-0	U-0	R/W-0	R-0	R-0	R-0	R-0	R-0
—	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0
bit 15							bit 8

R-0	R-1	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CRCFUL	CRCMPT	—	CRCGO	PLEN3	PLEN2	PLEN1	PLEN0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-14 **未实现**：读为 0

bit 13 **CSIDL**：空闲模式 CRC 停止位
1 = 当器件进入空闲模式时，模块不再继续工作
0 = 在空闲模式下模块继续工作

bit 12-8 **VWORD4:VWORD0**：指针值位
指出 FIFO 中有效字的个数。当 PLEN3:PLEN0 > 7 时，最大值为 8，
而当 PLEN3:PLEN0 ≤ 7 时，最大值为 16。

bit 7 **CRCFUL**：FIFO 满位
1 = FIFO 满
0 = FIFO 未满

bit 6 **CRCMPT**：FIFO 空位
1 = FIFO 为空
0 = FIFO 不空

bit 5 **未实现**：读为 0

bit 4 **CRCGO**：CRC 启动位
1 = 启动 CRC 串行移位器
0 = 关闭 CRC 串行移位器

bit 3-0 **PLEN3:PLEN0**：多项式长度位
表示将要生成的多项式的长度减 1。

寄存器 19-2： CRCXOR：CRC 异或多项式寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X15	X14	X13	X12	X11	X10	X9	X8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
X7	X6	X5	X4	X3	X2	X1	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-1 **X15:X1：**多项式的项 X^n 的异或操作使能位

bit 0 **未实现：**读为 0

 超前信息

20.0 10 位高速 A/D 转换器

> **注：** 本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的 **“第 17 章 10 位 A/D 转换器”**（DS39705A_CN）。

10 位 A/D 转换器具有以下主要特性：

- 逐次逼近（Successive Approximation，SAR）转换
- 转换速度最高可达 500 ksps
- 16 个模拟输入引脚
- 外部参考电压输入引脚
- 内部带隙参考输入
- 自动通道扫描模式
- 可选转换触发源
- 16 字转换结果缓冲器
- 可选缓冲器填充模式
- 四种结果对齐选项
- 可在 CPU 休眠和空闲模式下继续工作

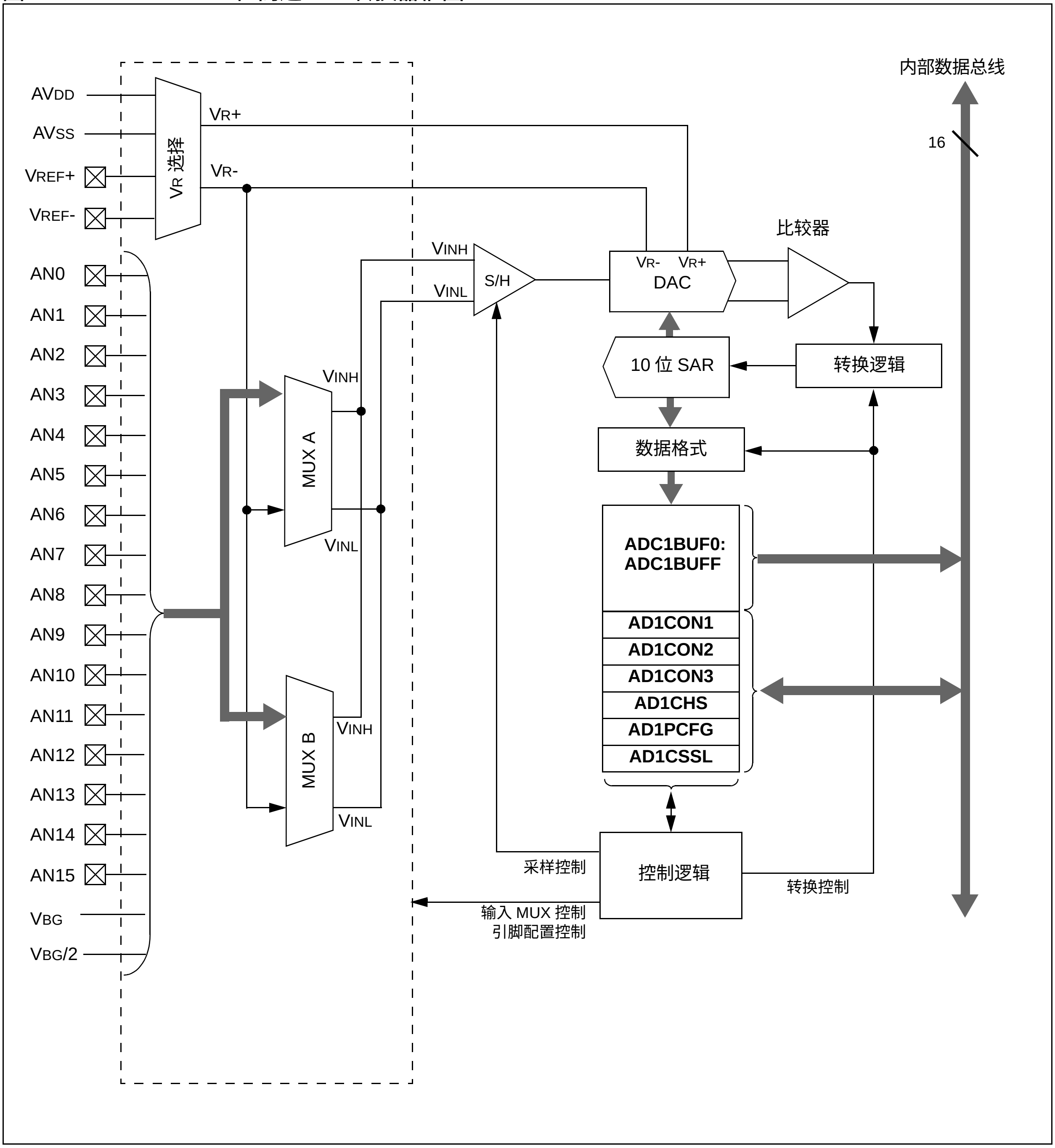
在全部 PIC24FJ256GA110 系列器件中， 10 位 A/D 转换器有 16 个模拟输入引脚，指定为从 AN0 至 AN15。此外，有两个可用于外部参考电压连接的模拟输入引脚（VREF+ 和 VREF-）。这两个参考电压输入引脚可以和其他模拟输入引脚复用。

图 20-1 为 A/D 转换器的框图。

要执行一个 A/D 转换：

1. 配置 A/D 模块：
   a） 选择端口引脚作为模拟输入引脚（AD1PCFG<15:0>）。
   b） 选择参考电压源以匹配模拟输入的预期范围（AD1CON2<15:13>）。
   c） 选择模拟转换时钟以便使期望的数据速率与处理器时钟匹配 （AD1CON3<7:0>）。
   d） 选择适当的采样 / 转换序列 （AD1CON1<7:5> 和 AD1CON3<12:8>）。
   e） 选择转换结果在缓冲器中的存储方式（AD1CON1<9:8>）。
   f） 选择中断频率 （AD1CON2<5:2>）。
   g） 启动 A/D 模块 （AD1CON1<15>）。
2. 配置 A/D 中断 （如果需要）：
   a） 清零 AD1IF 位。
   b） 选择 A/D 中断优先级。

图 20-1： 10 位高速 A/D 转换器框图



 超前信息

寄存器 20-1： AD1CON1：A/D 控制寄存器 1

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON(1)	—	ADSIDL	—	—	—	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0, HCS	R/W-0, HCS
SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE
bit 7							bit 0

图注：	HCS = 可由硬件清零 / 置 1 的位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **ADON：**A/D 工作模式位 (1)
1 = A/D 转换器模块正在工作
0 = A/D 转换器关闭

bit 14 **未实现：**读为 0

bit 13 **ADSIDL：**空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 模块在空闲模式下继续工作

bit 12-10 **未实现：**读为 0

bit 9-8 **FORM1:FORM0：**数据输出格式位
11 = 有符号小数 （sddd dddd dd00 0000）
10 = 小数 （dddd dddd dd00 0000）
01 = 有符号整数 （ssss sssd dddd dddd）
00 = 整数 （0000 00dd dddd dddd）

bit 7-5 **SSRC2:SSRC0：**转换触发源选择位
111 = 内部计数器结束采样并启动转换 （自动转换）
110 = 保留
101 = 保留
100 = CTMU 事件结束采样并启动转换
011 = Timer5 比较结束采样并启动转换
010 = Timer3 比较结束采样并启动转换
001 = INT0 引脚的有效跳变沿结束采样并启动转换
000 = SAMP 位清零结束采样并启动转换

bit 4-3 **未实现：**读为 0

bit 2 **ASAM：**A/D 采样自动启动位
1 = 最后一次转换结束后立即开始采样。 SAMP 位自动置 1。
0 = SAMP 位置 1 时开始采样

bit 1 **SAMP：**A/D 采样使能位
1 = A/D 采样 / 保持放大器正在采样输入
0 = A/D 采样 / 保持放大器正在保持

bit 0 **DONE：**A/D 转换状态位
1 = A/D 转换完成
0 = A/D 转换未完成

注 1： 一旦 ADON 位清零，ADC1BUFx 寄存器将不能保留其中的值。禁止该模块之前，从缓冲器中读出转换值。

 超前信息

寄存器 20-2： AD1CON2：A/D 控制寄存器 2

R/W-0	R/W-0	R/W-0	r-0	U-0	R/W-0	U-0	U-0
VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—
bit 15							bit 8

R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS
bit 7							bit 0

图注：		U = 未实现位，读为 0	
R = 可读位	W = 可写位	r = 保留位	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-13 **VCFG2:VCFG0**：参考电压配置位

VCFG2:VCFG0	VR+	VR-
000	AVDD	AVSS
001	外部 VREF+ 引脚	AVSS
010	AVDD	外部 VREF- 引脚
011	外部 VREF+ 引脚	外部 VREF- 引脚
1xx	AVDD	AVSS

bit 12 **保留**：保持为 0

bit 11 **未实现**：读为 0

bit 10 **CSCNA**：选择是否在使用采样多路开关 A 时扫描 CH0+ S/H 输入的位
1 = 扫描输入
0 = 不扫描输入

bit 9-8 **未实现**：读为 0

bit 7 **BUFS**：缓冲器填充状态位 （只在 BUFM = 1 时有效）
1 = A/D 当前正在填充缓冲器 08-0F，用户应该访问 00-07 中的数据
0 = A/D 当前正在填充缓冲器 00-07，用户应该访问 08-0F 中的数据

bit 6 **未实现**：读为 0

bit 5-2 **SMPI3:SMPI0**：每次中断的采样 / 转换序列执行次数的选择位
1111 = 每完成 16 次采样 / 转换序列后产生中断
1110 = 每完成 15 次采样 / 转换序列后产生中断
.....
0001 = 每完成 2 次采样 / 转换产生一次中断
0000 = 每完成 1 次采样 / 转换产生一次中断

bit 1 **BUFM**：缓冲器模式选择位
1 = 缓冲器配置为两个 8 字缓冲器 （ADC1BUFn<15:8> 和 ADC1BUFn<7:0>）
0 = 缓冲器配置为一个 16 字缓冲器 （ADC1BUFn<15:0>）

bit 0 **ALTS**：交替输入采样模式选择位
1 = 使用 MUX A 输入多路开关设置进行第一次采样，然后交替使用 MUX B 和 MUX A 输入多路开关设置进行所有后续采样
0 = 总是使用 MUX A 输入多路开关设置

寄存器 20-3： AD1CON3：A/D 控制寄存器 3

R/W-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
bit 7							bit 0

图注：		r = 保留位	
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **ADRC**：A/D 转换时钟源位
1 = A/D 内部 RC 时钟
0 = 时钟由系统时钟产生

bit 14-13 **保留**：保持为 0

bit 12-8 **SAMC4:SAMC0**：自动采样时间位
11111 = 31 TAD
…
00001 = 1 TAD
00000 = 0 TAD （不推荐）

bit 7-0 **ADCS7:ADCS0**：A/D 转换时钟选择位
11111111 = 256 TCY
…
00000001 = 2 TCY
00000000 = TCY

寄存器 20-4： AD1CHS0：A/D 输入选择寄存器

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	CH0SB4[1]	CH0SB3[1]	CH0SB2[1]	CH0SB1[1]	CH0SB0[1]
bit 15							bit 8

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **CH0NB：**采样多路开关 B 设置的通道 0 的反相输入选择位
1 = 通道 0 反相输入为 AN1
0 = 通道 0 反相输入为 VR-

bit 14-13 **未实现：**读为 0

bit 12-8 **CH0SB4:CH0SB0：**采样多路开关 B 设置的通道 0 的同相输入选择位 [1]
11111 = 通道 0 的同相输入为内部参考带隙 （VBG）
11110 = 通道 0 的同相输入为 VBG/2
01111 = 通道 0 的同相输入为 AN15
01110 = 通道 0 的同相输入为 AN14
01101 = 通道 0 的同相输入为 AN13
01100 = 通道 0 的同相输入为 AN12
01011 = 通道 0 的同相输入为 AN11
01010 = 通道 0 的同相输入为 AN10
01001 = 通道 0 的同相输入为 AN9
01000 = 通道 0 的同相输入为 AN8
00111 = 通道 0 的同相输入为 AN7
00110 = 通道 0 的同相输入为 AN6
00101 = 通道 0 的同相输入为 AN5
00100 = 通道 0 的同相输入为 AN4
00011 = 通道 0 的同相输入为 AN3
00010 = 通道 0 的同相输入为 AN2
00001 = 通道 0 的同相输入为 AN1
00000 = 通道 0 的同相输入为 AN0

bit 7 **CH0NA：**采样多路开关 A 设置的通道 0 的反相输入选择位
1 = 通道 0 的反相输入为 AN1
0 = 通道 0 的反相输入为 VR-

bit 6-5 **未实现：**读为 0

bit 4-0 **CH0SA4:CH0SA0：**采样多路开关 A 设置的通道 0 的同相输入选择位
实现的组合与 CHOSB4:CHOSB0 （同上）相同。

注 1： 从 10000 到 11101 的组合未实现；不可用。

寄存器 20-5： AD1PCFG：A/D 端口配置寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-0 **PCFG15:PCFG0**：模拟输入引脚配置控制位
1 = 相应模拟通道的引脚被配置为数字模式；使能 I/O 端口读操作
0 = 引脚被配置为模拟模式；禁止 I/O 端口读操作， A/D 对引脚电压进行采样

寄存器 20-6： AD1CSSL：A/D 输入扫描选择寄存器 （低位字）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-0 **CSSL15:CSSL0**：A/D 输入引脚扫描选择位
1 = 为输入扫描选择的对应模拟通道
0 = 被输入扫描忽略的对应模拟通道

寄存器 20-7： AD1CSSH：A/D 输入扫描选择寄存器 （高位字）

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	CSSL17	CSSL16
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-2 **未实现：**读为 0

bit 1 **CSSL17：**A/D 输入带隙扫描选择位
1 = 为输入扫描选择的内部带隙 （V_{BG}）通道
0 = 被输入扫描忽略的模拟通道

bit 0 **CSSL16：**A/D 输入半带隙扫描选择位
1 = 为输入扫描选择的内部 $V_{BG}/2$ 通道
0 = 被输入扫描忽略的模拟通道

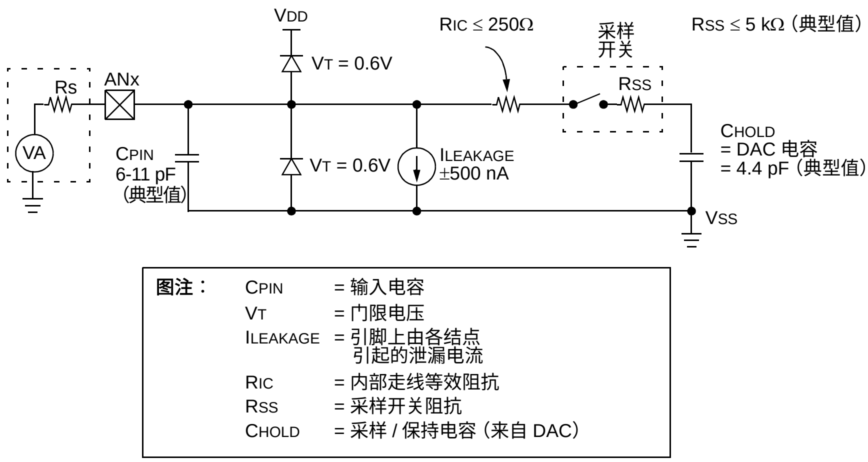
公式 20-1： A/D 转换时钟周期 (1)

$$T_{AD} = T_{CY} \cdot (ADCS + 1)$$

$$ADCS = \frac{T_{AD}}{T_{CY}} - 1$$

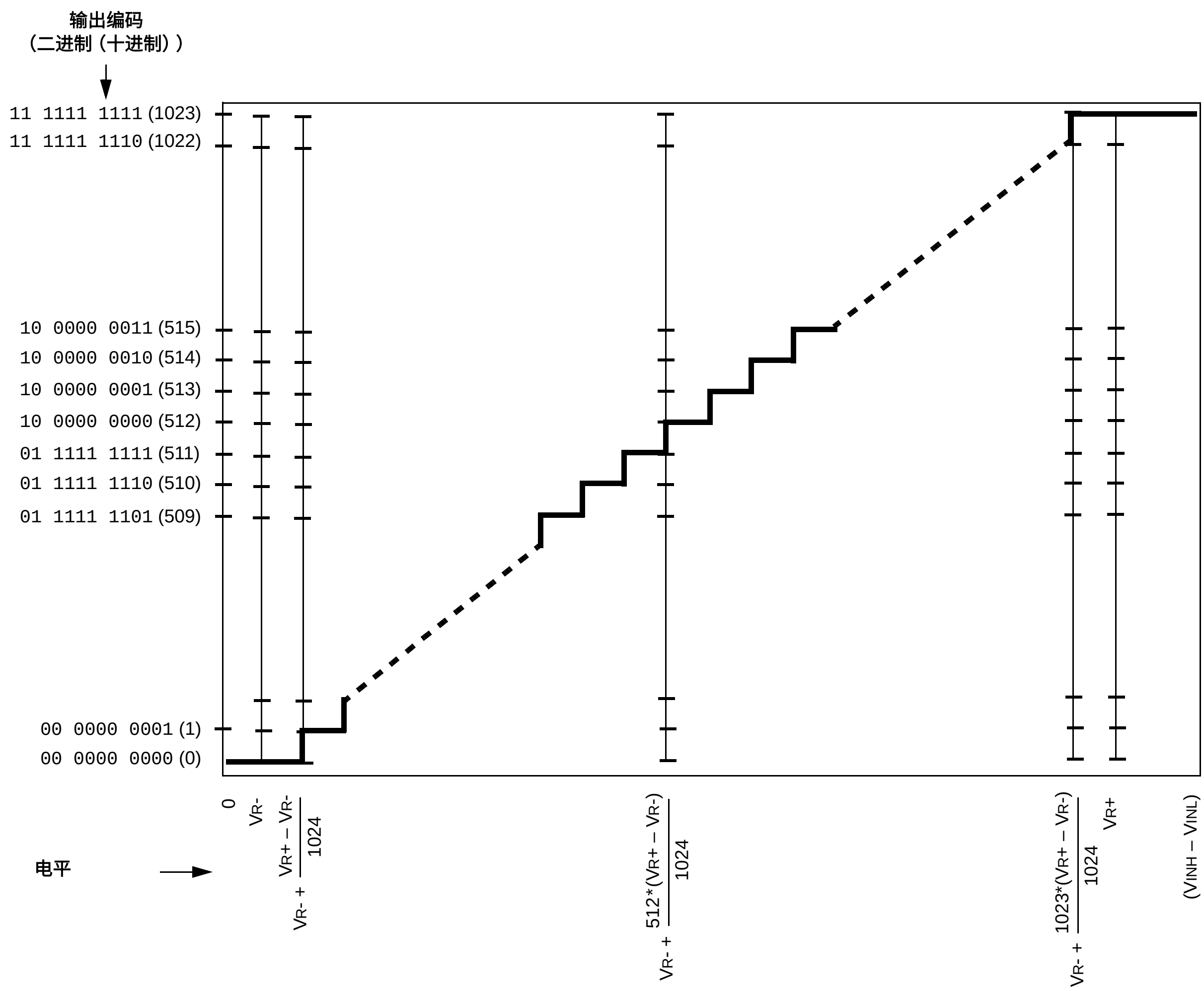
注 1： 基于 $T_{CY} = 2 * T_{OSC}$ ；禁止打盹模式和 PLL。

图 20-2： 10 位 A/D 转换器模拟输入模型



注：CPIN 值取决于器件封装，未经测试。 CPIN 的影响可忽略 （在 Rs ≤ 5 kΩ 的情况下）。

图 20-3： A/D 传递函数



注：

21.0 三比较器模块

> **注：** 本数据手册总结了 PIC24F 系列器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》中的相关章节。

三比较器模块提供了三个双输入的比较器。至比较器的输入可以配置为使用 4 个外部模拟输入之一，而且参考电压输入可以来自内部带隙参考的 2 分频 （$V_{BG}/2$）信号或比较器参考电压发生器。

比较器输出可能直接连接到 CxOUT 引脚。当对应的 COE 等于 1 时，I/O 引脚逻辑使比较器的不同步输出能够在引脚上使用。

图 21-1 所示为模块的简化框图。各种可能的比较器配置图如图 21-2 所示。

每个比较器都有自己的控制寄存器，CMxCON（见寄存器 21-1），用来使能和配置其操作。 CMSTAT 寄存器（见寄存器 21-2）提供了所有三个比较器的事件状态和输出。

图 21-1： 三比较器模块框图

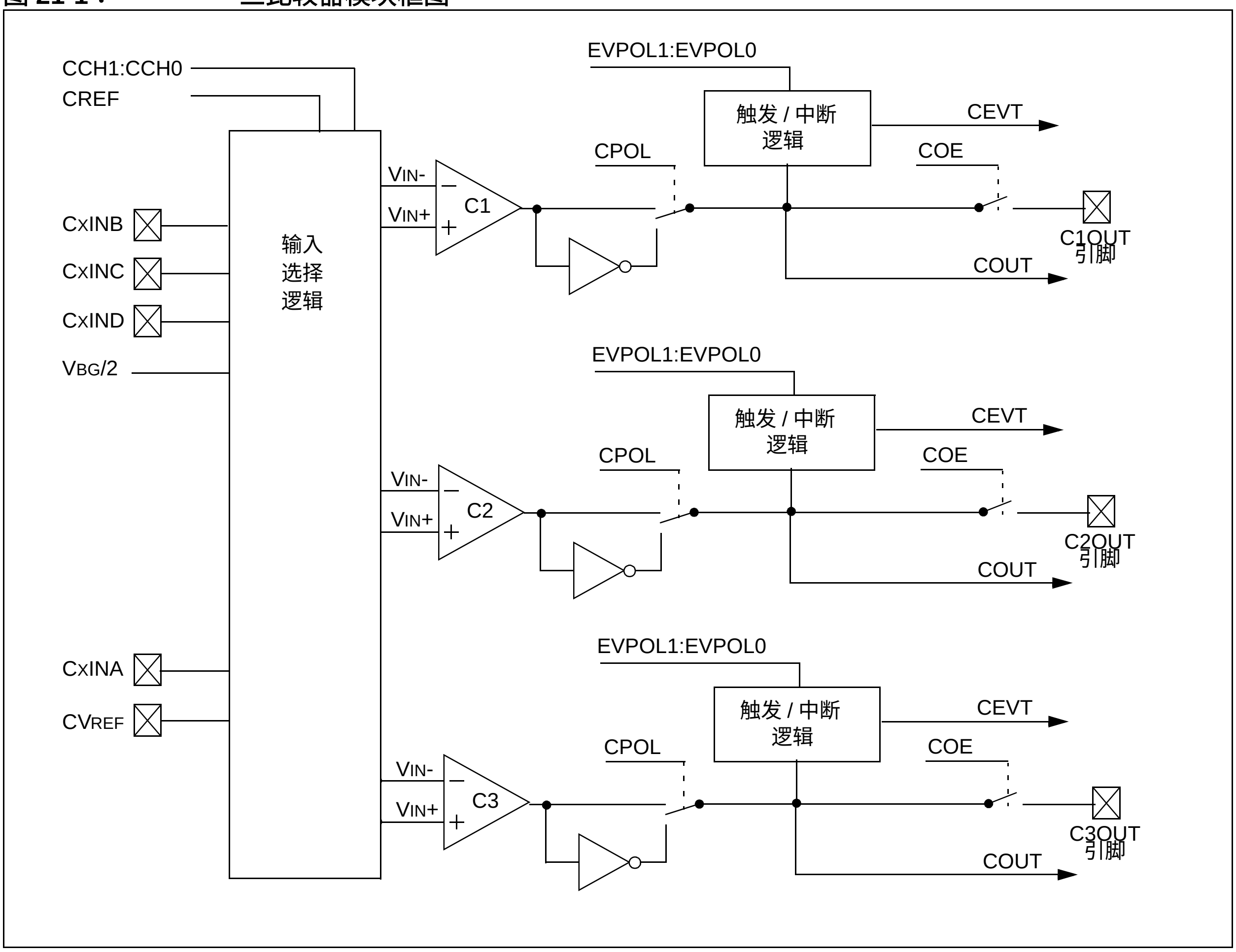
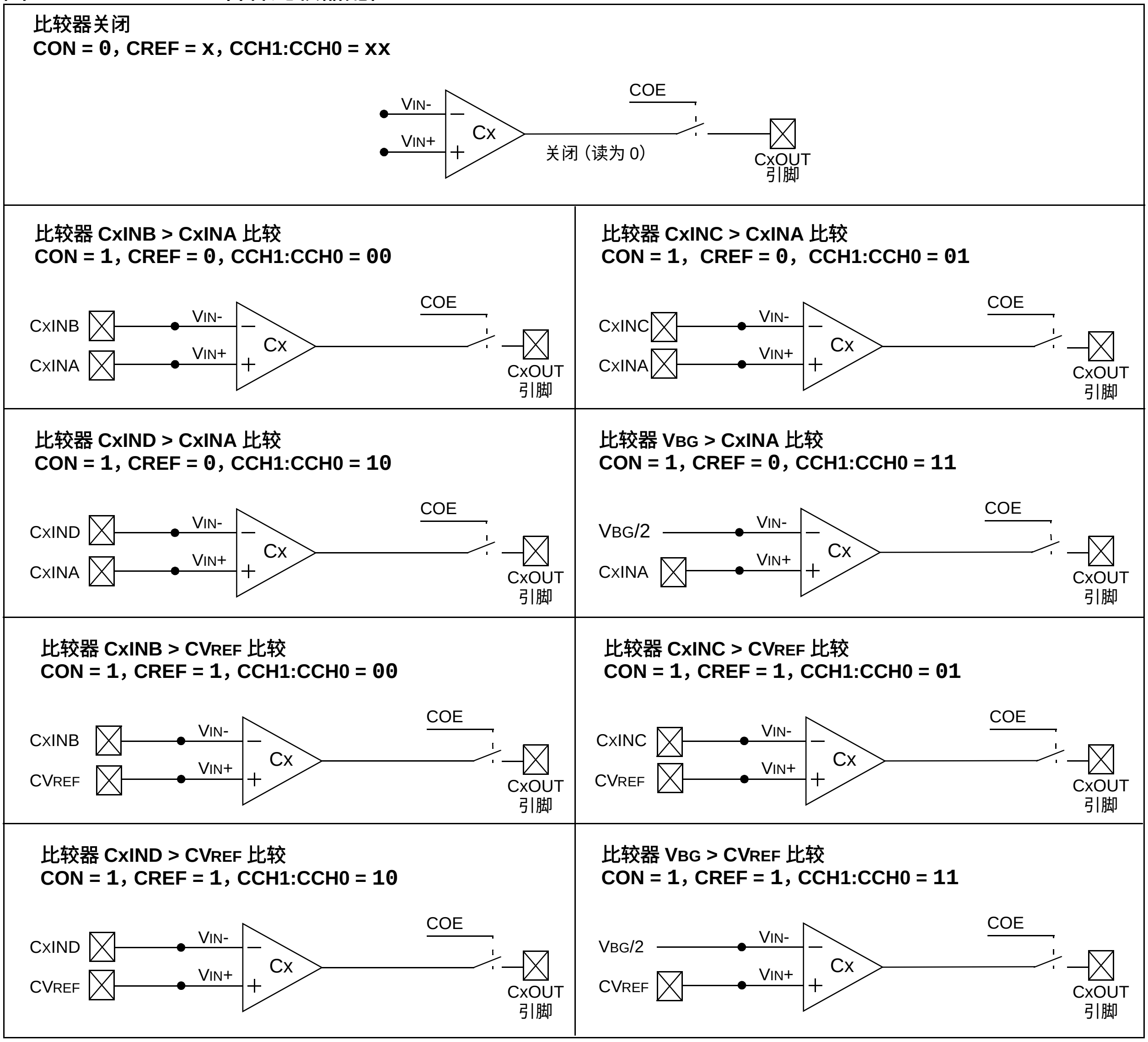


图 21-2： 各种比较器配置



寄存器 21-1： CMxCON：比较器 x 控制寄存器 （比较器 1 到 3）

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R-0
CON	COE	CPOL	—	—	—	CEVT	COUT
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **CON**：比较器使能位
1 = 比较器被使能
0 = 比较器被禁止

bit 14 **COE**：比较器输出使能位
1 = 在 CxOUT 引脚上有比较器输出
0 = 比较器输出无外部连接

bit 13 **CPOL**：比较器输出极性选择位
1 = 比较器输出反相
0 = 比较器输出不反相

bit 12-10 **未实现**：读为 0

bit 9 **CEVT**：比较器事件位
1 = 发生了由 EVPOL1:EVPOL0 定义的比较器事件；随后的触发和中断被禁止，直到该位清零
0 = 未发生比较器事件

bit 8 **COUT**：比较器输出位
当 CPOL = 0 时：
1 = VIN+ > VIN-
0 = VIN+ < VIN-
当 CPOL = 1 时：
1 = VIN+ < VIN-
0 = VIN+ > VIN-

bit 7-6 **EVPOL1:EVPOL0**：触发 / 事件 / 中断极性选择位
11 = 由比较器输出 （当 CEVT = 0 时）的任何变化产生触发 / 事件 / 中断
10 = 由比较器输出的跳变产生触发 / 事件 / 中断：
如果 CPOL = 0 （极性不变）：
仅限从高到低跳变。
如果 CPOL = 1 （极性反转）：
仅限从低到高跳变。
01 = 由比较器输出的跳变产生触发 / 事件 / 中断：
如果 CPOL = 0 （极性不变）：
仅限从低到高跳变。
如果 CPOL = 1 （极性反转）：
仅限从高到低跳变。
00 = 禁止产生触发 / 事件 / 中断

bit 5 **未实现**：读为 0

寄存器 21-1： CMxCON：比较器 x 控制寄存器 （比较器 1 到 3） （续）

bit 4 **CREF**：比较器参考电压选择位 （同相输入）
1 = 同相输入端连接到内部 CVREF 电压
0 = 同相输入端连接到 CxINA 引脚

bit 3-2 **未实现**：读为 0

bit 1-0 **CCH1:CCH0**：比较器通道选择位
11 = 比较器的反相输入端连接到 VBG/2
10 = 比较器的反相输入端连接到 CxIND 引脚
01 = 比较器的反相输入端连接到 CxINC 引脚
00 = 比较器的反相输入端连接到 CxINB 引脚

寄存器 21-2： CMSTAT：比较器模块状态寄存器

R/W-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
CMIDL	—	—	—	—	C3EVT	C2EVT	C1EVT
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	C3OUT	C2OUT	C1OUT
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **CMIDL**：空闲模式下比较器停止位
1 = 器件进入空闲模式时，所有比较器都停止工作
0 = 在空闲模式下所有被使能的比较器都继续工作

bit 14-11 **未实现**：读为 0

bit 10 **C3EVT**：比较器 3 事件状态位 （只读）
显示比较器 3 的当前事件状态 （CM3CON<9>）。

bit 9 **C2EVT**：比较器 2 事件状态位 （只读）
显示比较器 2 的当前事件状态 （CM2CON<9>）。

bit 8 **C1EVT**：比较器 1 事件状态位 （只读）
显示比较器 1 的当前事件状态 （CM1CON<9>）。

bit 7-3 **未实现**：读为 0

bit 2 **C3OUT**：比较器 3 输出状态位 （只读）
显示比较器 3 的当前输出状态 （CM3CON<8>）。

bit 1 **C2OUT**：比较器 2 输出状态位 （只读）
显示比较器 2 的当前输出状态 （CM2CON<8>）。

bit 0 **C1OUT**：比较器 1 输出状态位 （只读）
显示比较器 1 的当前输出状态 （CM1CON<8>）。

22.0 比较器参考电压

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的**“第 20 章 比较器参考电压模块”**（DS39709A_CN）。

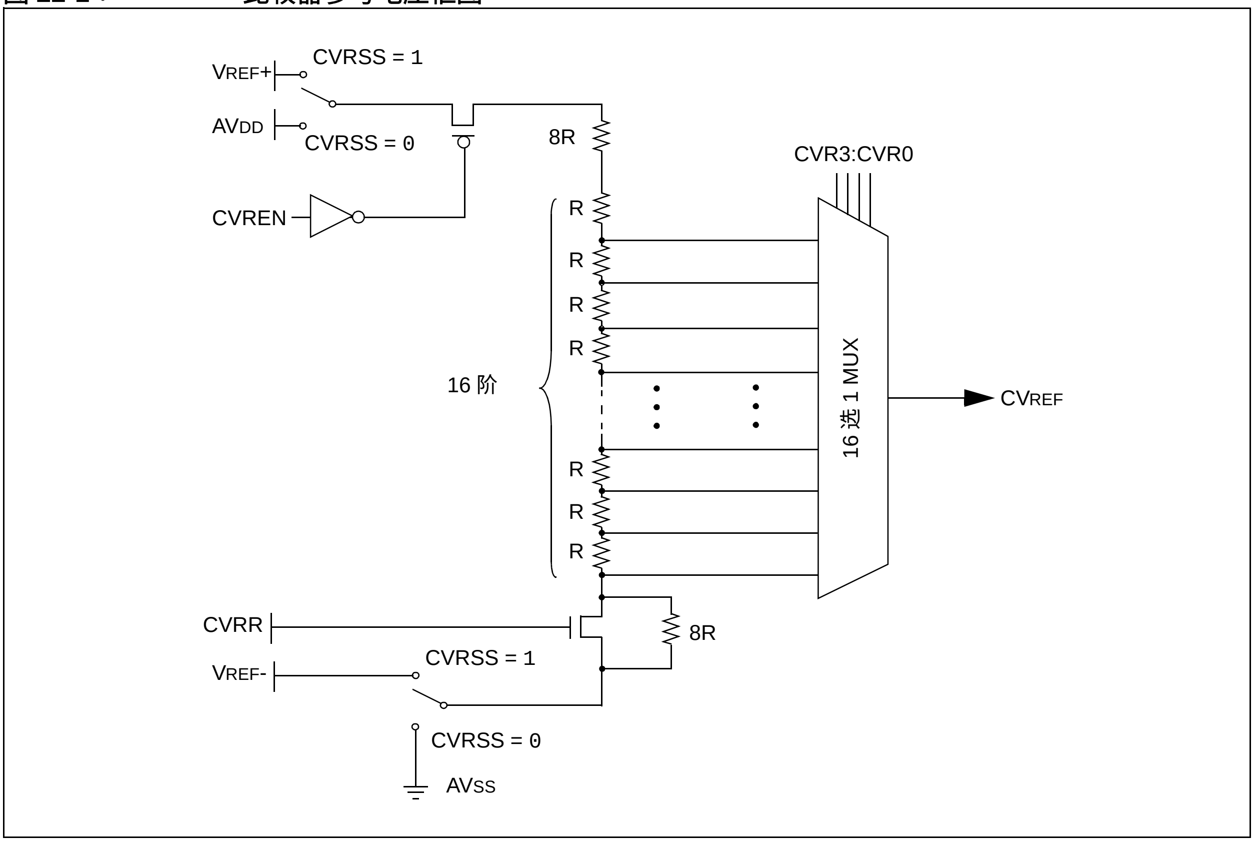
22.1 配置比较器参考电压

参考电压模块由 CVRCON 寄存器（寄存器 22-1）控制。比较器参考电压提供两种范围的输出电压，每种范围都具有 16 种不同的电平。CVRR 位（CVRCON<5>）选择要使用的电压范围。这两种范围的主要区别在于 CVREF 选择位（CVR3:CVR0）所选的步长，其中一个范围提供更高的分辨率。

比较器参考电压模块的电压源来自于 VDD 和 VSS 或外部 VREF+ 和 VREF-。由 CVRSS 位（CVRCON<4>）选择电压源。

在改变 CVREF输出时必须考虑到比较器参考电压的稳定时间。

图 22-1： 比较器参考电压框图



寄存器 22-1： CVRCON：比较器参考电压控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-8 **未实现：**读为 0

bit 7 **CVREN：**比较器参考电压使能位
1 = CVREF 电路上电
0 = CVREF 电路断电

bit 6 **CVROE：**比较器 VREF 输出使能位
1 = CVREF 电压为 CVREF 引脚的输出
0 = CVREF 电压与 CVREF 引脚断开

bit 5 **CVRR：**比较器 VREF 范围选择位
1 = CVRSRC 范围应从 0 到 0.625 CVRSRC，步长为 CVRSRC/24
0 = CVRSRC 范围应从 0.25 到 0.719 CVRSRC，步长为 CVRSRC/32

bit 4 **CVRSS：**比较器 VREF 源选择位
1 = 比较器参考源 CVRSRC = VREF+ – VREF-
0 = 比较器参考源 CVRSRC = AVDD – AVSS

bit 3-0 **CVR3:CVR0：**比较器 VREF 值选择位 （$0 \leq CVR3:CVR0 \leq 15$）
当 CVRR = 1 时：
$CV_{REF} = (CVR<3:0>/ 24) \cdot (CV_{RSRC})$
当 CVRR = 0 时：
$CV_{REF} = 1/4 \cdot (CV_{RSRC}) + (CVR<3:0>/32) \cdot (CV_{RSRC})$

23.0 充电时间测量单元 （CTMU）

注：	本数据手册总结了 PIC24F 系列器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》中的相关章节。

充电时间测量单元是一个灵活的模拟模块，它可提供对脉冲源输出脉冲时间差的精确测量，并生成异步脉冲。模块的主要特性包括：

- 4 个边沿输入触发源
- 对每个边沿源进行极性控制
- 对边沿序列进行控制
- 控制对边沿的响应
- 时间测量的分辨率为 1 纳秒
- 适合电容测量的精确电流源

与其他片上模拟模块一起，CTMU 可用于精确地测量时间、电容值、电容值的相关变化或生成与系统时钟操作无关的输出脉冲。CTMU 模块是与基于容性的传感器接口的理想选择。

可通过两个寄存器 CTMUCON 和 CTMUICON 控制 CTMU。 CTMUCON 使能 CTMU 模块，并控制边沿源选择、边沿源极性选择和边沿序列。而 CTMUICON 寄存器控制电流源的选择和微调。

23.1 测量电容

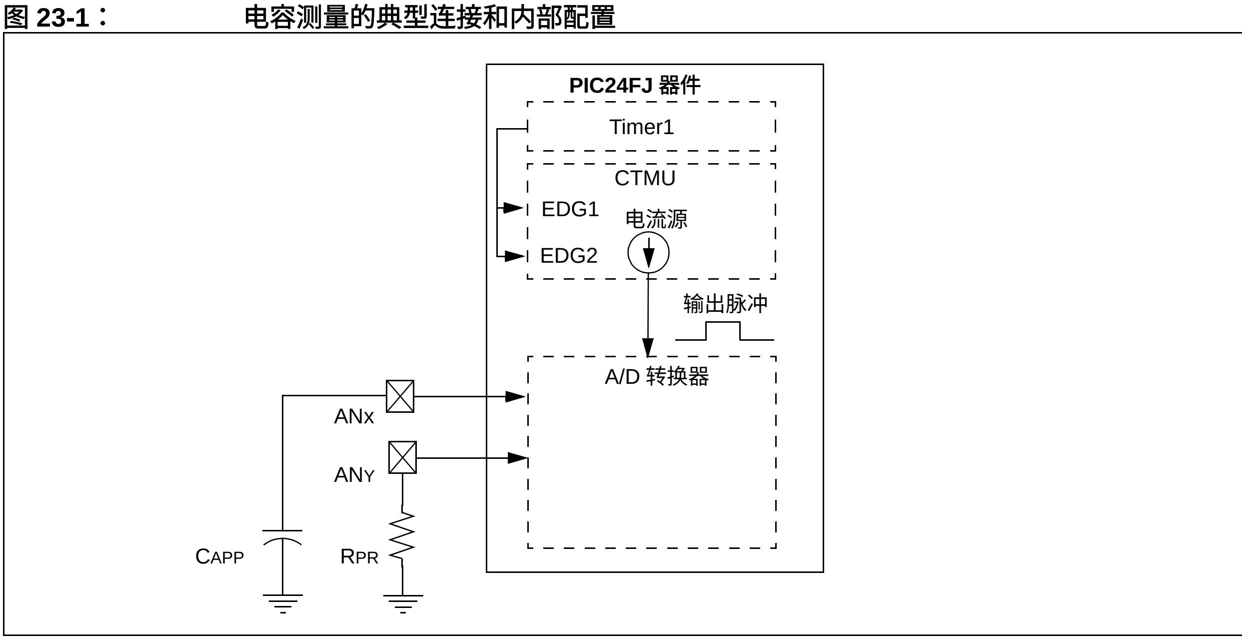
CTMU 模块通过生成脉宽与两个独立输入通道上的边沿事件之间的时间相等的输出脉冲来测量电容。可以从以下四个源中选择到这两个输入通道上的脉冲边沿事件：两个内部外设模块 （OC1 和 Timer1） ；两个外部引脚 （CTEDG1 和 CTEDG2）。脉冲和模块的精确电流源一起可用来根据下列关系式计算电容值。

$$C = I \cdot \frac{dV}{dT}$$

为了测量电容，A/D 转换器在 CTMU 输出脉冲后对输入通道之一上的外部电容 （CAPP）进行采样。由第二个 A/D 通道上的精确电阻 （RPR）提供电流源校准。脉冲结束后，转换器会确定电容上的电压值。由应用程序使用软件执行电容值的实际计算。

图23-1显示用于电容测量的外部连接，以及应用中CTMU 和 A/D 模块的连接方式。该示例还显示来自 Timer1 的边沿事件，但使用外部边沿源的其他配置也是可能的。《PIC24F 系列参考手册》中提供了关于测量 CTMU 模块的电容值和时间的详细讨论。

图 23-1： 电容测量的典型连接和内部配置



23.2 测量时间

对脉宽进行时间测量可采用与使用 A/D 模块的内部电容（C_{AD}）和精度电阻进行电流校准类似的方法进行。图 23-2 显示了用于时间测量的外部连接，以及应用中 CTMU 和 A/D 模块的连接方式。该示例还显示了两个来自外部 CTEDG 引脚的边沿事件，但使用外部边沿源的其他配置也是可能的。《PIC24F 系列参考手册》中提供了关于测量 CTMU 模块的电容值和时间的详细讨论。

23.3 脉冲的产生和延时

CTMU 模块也能产生边沿与器件的系统时钟不同步的输出脉冲。具体而言，该模块可以产生距离边沿事件输入到模块的延时可编程的脉冲。

在通过 TGEN 位（CTMUCON<12>）置 1 配置模块具有脉冲生成延时功能时，内部电流源连接到比较器 2 的 B 输入引脚上。电容（C_{DELAY}）连接到比较器 2 的引脚C2INB上，比较器参考电压CV_{REF}连接到引脚C2INA上。CV_{REF} 稍后被配置为指定的跳变点。当检测到边沿事件时，模块开始对电容 C_{DELAY} 充电。当 C_{DELAY} 被充电到大于 CV_{REF} 跳变点时，从 CTPLS 上输出脉冲。脉冲延时的长度由 C_{DELAY} 的值和 CV_{REF} 跳变点确定。

图 23-3 显示用于生成脉冲的外部连接，以及所需的不同模拟模块之间的关系。当 CTEDG1 作为输入脉冲源时，其余选项可用。《PIC24F 系列参考手册》中提供了关于 CTMU 模块生成脉冲的详细讨论。

图 23-2： 时间测量的典型连接和内部配置

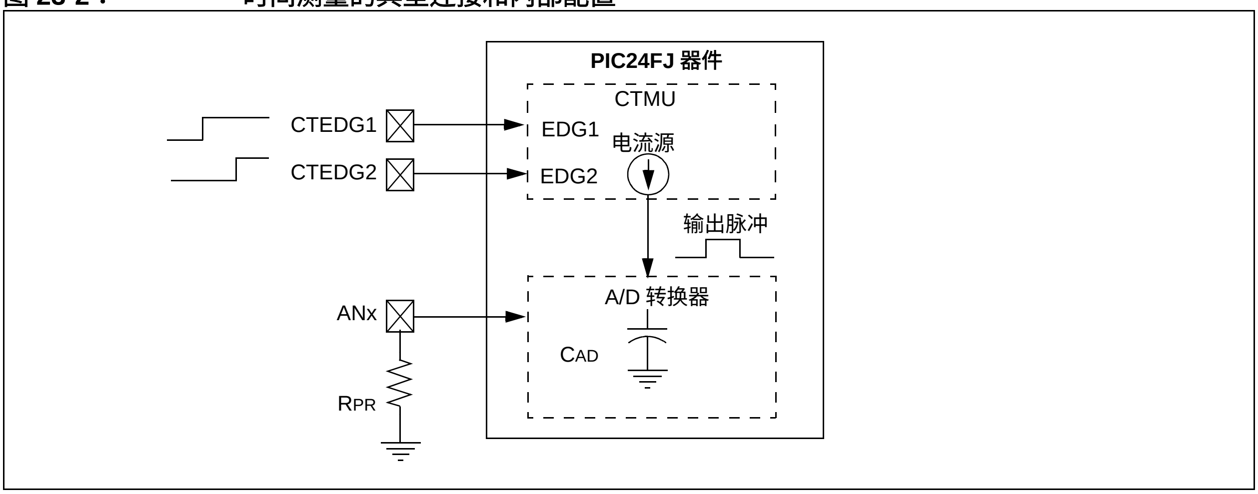
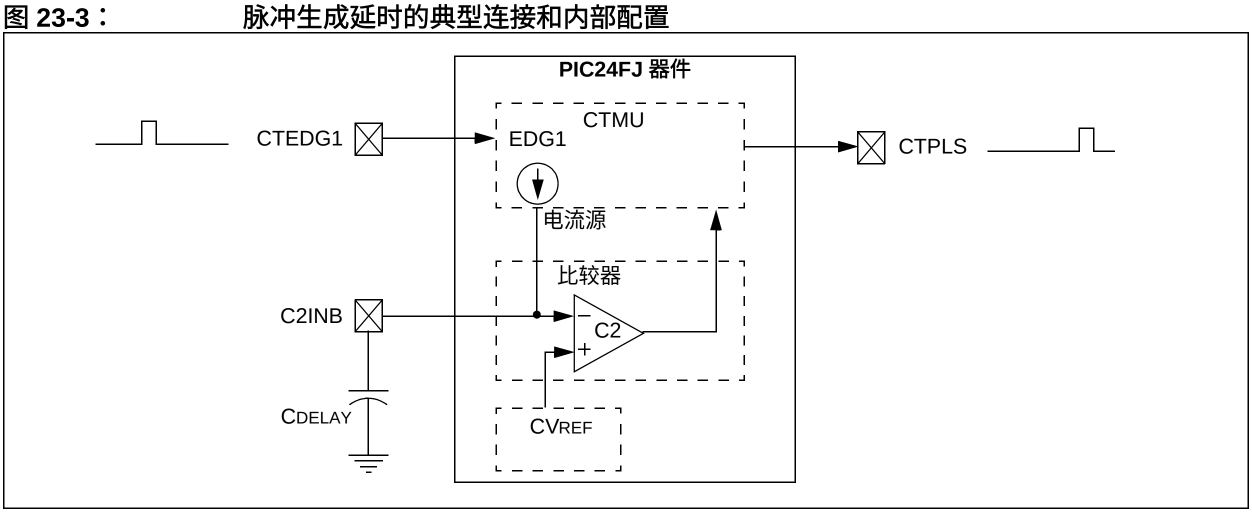


图 23-3： 脉冲生成延时的典型连接和内部配置



寄存器 23-1： CTMUCON：CTMU 控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15 **CTMUEN**：CTMU 使能位
1 = 使能模块
0 = 禁止模块

bit 14 **未实现**：读为 0

bit 13 **CTMUSIDL**：空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 模块在空闲模式下继续工作

bit 12 **TGEN**：时间生成使能位
1 = 使能边沿延时生成
0 = 禁止边沿延时生成

bit 10 **EDGEN**：边沿使能位
1 = 不阻止边沿
0 = 阻止边沿

bit 10 **EDGSEQEN**：边沿序列使能位
1 = 边沿 1 事件必须在边沿 2 事件发生之前发生
0 = 无需边沿序列

bit 9 **IDISSEN**：模拟电流源控制位
1 = 模拟电流源输出接地
0 = 模拟电流源输出不接地

bit 8 **CTTRIG**：触发器控制位
1 = 使能触发器输出
0 = 禁止触发器输出

bit 7 **EDG2POL**：边沿 2 极性选择位
1 = 对边沿 2 进行编程以响应上升沿
0 = 对边沿 2 进行编程以响应下降沿

bit 6-5 **EDG2SEL1:EDG2SEL0**：边沿 2 源选择位
11 = CTED1 引脚
10 = CTED2 引脚
01 = OC1 模块
00 = Timer1 模块

bit 4 **EDG1POL**：边沿 1 极性选择位
1 = 对边沿 1 进行编程以响应上升沿
0 = 对边沿 1 进行编程以响应下降沿

寄存器 23-1： CTMUCON：CTMU 控制寄存器 （续）

bit 3-2 **EDG1SEL1:EDG1SEL0：**边沿 1 源选择位
11 = CTED1 引脚
10 = CTED2 引脚
01 = OC1 模块
00 = Timer1 模块

bit 1 **EDG2STAT：**边沿 2 状态位
1 = 发生边沿 2 事件
0 = 未发生边沿 2 事件

bit 0 **EDG1STAT：**边沿 1 状态位
1 = 发生边沿 1 事件
0 = 未发生边沿 1 事件

寄存器 23-2： CTMUICON：CTMU 电流控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-10 **ITRIM5:ITRIM0：**电流源微调位
011111 = 对标称电流的最大正向调整
011110
...
000001 = 对标称电流的最小正向调整
000000 = IRNG1:IRNG0 指定的标称电流输出
111111 = 对标称电流的最小负向调整
...
100010
100001 = 对标称电流的最大负向调整

bit 9-8 **IRNG1:IRNG0：**电流源范围选择位
11 = 100 × 基本电流
10 = 10 × 基本电流
01 = 基本电流 （标称值为 0.55 μA）
00 = 禁止电流源

bit 7-0 **未实现：**读为 0

24.0 特殊功能

> **注：** 本数据手册总结了 PIC24F 系列器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》中的下列章节：
> - **第 9 章 " 看门狗定时器 （WDT） "** （DS39697A_CN）
> - **第 32 章 " 高级器件集成 "** （DS39719A_CN）
> - **第 33 章 " 编程和诊断 "** （DS39716A_CN）

PIC24FJ256GA110 系列器件包含的功能旨在最大限度地提高应用的灵活性和可靠性，并通过减去外部元件把成本降到最低。这些功能包括了：

- 灵活的配置
- 看门狗定时器 （WDT）
- 代码保护
- JTAG 边界扫描接口
- 在线串行编程
- 在线仿真

24.1 配置位

可以通过对配置位编程 （读为 `0`）或保留不编程 （读为 `1`）来选择不同的器件配置。这些配置位被映射到程序存储器中从 F8000h 开始的单元中。从寄存器 24-1 到寄存器 24-5 详细解释了各配置位的不同功能。

注意，地址F80000h超出了用户程序存储空间。事实上，它属于配置存储空间 （800000h-FFFFFFh），这一空间仅能通过表读和表写进行访问。

24.1.1 配置 PIC24FJ256GA110 系列器件的注意事项

在 PIC24FJ256GA110 系列器件中，配置字节以易失性存储方式实现。这就意味着在器件每次上电时都必须对配置数据进行编程。配置数据存储在片上程序存储空间顶部的三个字中，这些字被称为闪存配置字。表 24-1 显示了它们的具体位置。这些是实际器件配置位的紧凑表现形式，这些配置位实际分布在配置空间中的多个位置中。器件复位过程中，配置数据会自动从闪存配置字装入到相应的配置寄存器中。

> **注：** 所有类型的器件复位都会重新装载配置数据。

当为这些器件创建应用程序时，用户就该为配置数据分配特定的闪存配置字单元。此操作是为了确保在编译代码时不会把程序代码存储在该地址上。

程序存储器中的所有闪存配置字的高字节应始终为 " `1111 1111` "。这样使得当远程事件意外执行这些存储单元时将其作为 `NOP` 执行来执行。由于配置位不是在对应的单元内实现的，因此向这些单元写 " `1` " 不会影响器件工作。

> **注：** 在程序存储器末页中执行页擦除操作会清零闪存配置字并且使能代码保护。因此，应避免在程序存储器末页中执行页擦除操作。

表 24-1： PIC24FJ256GA110 系列器件的闪存配置字位置

器件	配置字地址		
	1	**2**	**3**
PIC24FJ128GA1	157FEh	157FC	157FA
PIC24FJ192GA1	20BFEh	20BFC	20BFA
PIC24FJ256GA1	2ABFEh	2ABFC	2ABFA

寄存器 24-1： CW1：闪存配置字 1

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

r-x	R/PO-1	R/PO-1	R/PO-1	R/PO-1	r-1	R/PO-1	R/PO-1
r	JTAGEN	GCP	GWRP	$\overline{\text{DEBUG}}$	r	ICS1	ICS0
bit 15							bit 8

R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
FWDTEN	WINDIS	—	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

图注：	r = 保留位		
R = 可读位	PO = 一次编程位	U = 未实现位，读为 0	
-n = 未对器件编程时的值		1 = 置 1	0 = 清零

bit 23-16 **未实现：**读为 1

bit 15 **保留：**未知值；编程为 0

bit 14 **JTAGEN：**JTAG 端口使能位 [(1)]
1 = 使能 JTAG 端口
0 = 禁止 JTAG 端口

bit 13 **GCP：**通用段程序存储器代码保护位
1 = 禁止代码保护
0 = 对整个程序存储空间使能代码保护

bit 12 **GWRP：**通用段代码闪存写保护位
1 = 允许写程序存储器
0 = 禁止写程序存储器

bit 11 **$\overline{\text{DEBUG}}$：**后台调试器使能位
1 = 器件复位到工作模式
0 = 器件复位到调试模式

bit 10 **保留：**始终保持为 1

bit 9-8 **ICS1:ICS0：**仿真器引脚位置选择位
11 = 仿真器功能与 PGEC1/PGED1 共用
10 = 仿真器功能与 PGEC2/PGED2 共用
01 = 仿真器功能与 PGEC3/PGED3 共用
00 = 保留；不使用

bit 7 **FWDTEN：**看门狗定时器使能位
1 = 使能看门狗定时器
0 = 禁止看门狗定时器

bit 6 **WINDIS：**窗口看门狗定时器禁止位
1 = 使能标准看门狗定时器
0 = 使能窗口看门狗定时器； FWDTEN 必须为 1

bit 5 **未实现：**读为 1

bit 4 **FWPSA：**WDT 预分频比选择位
1 = 预分频比为 1:128
0 = 预分频比为 1:32

注 1： 仅能使用在线串行编程 （ICSP™）修改 JTAGEN 位。在通过 JTAG 接口对器件进行编程时，无法修改 JTAGEN 位。

寄存器 24-1： CW1：闪存配置字 1 （续）

bit 3-0 **WDTPS3:WDTPS0**：看门狗定时器后分频比选择位
`1111` = 1:32,768
`1110` = 1:16,384
`1101` = 1:8,192
`1100` = 1:4,096
`1011` = 1:2,048
`1010` = 1:1,024
`1001` = 1:512
`1000` = 1:256
`0111` = 1:128
`0110` = 1:64
`0101` = 1:32
`0100` = 1:16
`0011` = 1:8
`0010` = 1:4
`0001` = 1:2
`0000` = 1:1

注 1： 仅能使用在线串行编程 （ICSP™）修改 JTAGEN 位。在通过 JTAG 接口对器件进行编程时，无法修改 JTAGEN 位。

寄存器 24-2： CW2：闪存配置字 2

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	U-1	U-1	U-1	r-0	R/PO-1	R/PO-1	R/PO-1
IESO	—	—	—	r	FNOSC2	FNOSC1	FNOSC0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1
FCKSM1	FCKSM0	OSCIOFCN	IOL1WAY	—	I2C2SEL[(1)]	POSCMD1	POSCMD0
bit 7							bit 0

图注：		r = 保留位	
R = 可读位	PO = 一次编程位	U = 未实现位，读为 0	
-n = 未对器件编程时的值		1 = 置 1	0 = 清零

bit 23-16 **未实现：**读为 1

bit 15 **IESO：**内外部时钟源切换位
- 1 = 使能 IESO 模式 （双速启动）
- 0 = 禁止 IESO 模式 （双速启动）

bit 14-12 **未实现：**读为 1

bit 11 **保留：**始终保持为 0

bit 10-8 **FNOSC2:FNOSC0：**初始振荡器选择位
- 111 = 带后分频器的快速 RC 振荡器 （FRCDIV）
- 110 = 保留
- 101 = 低功耗 RC 振荡器 （LPRC）
- 100 = 辅助振荡器 （SOSC）
- 011 = 带 PLL 模块的主振荡器 （XTPLL、 HSPLL 和 ECPLL）
- 010 = 主振荡器 （XT、 HS 和 EC）
- 001 = 带后分频器和 PLL 模块的快速 RC 振荡器 （FRCPLL）
- 000 = 快速 RC 振荡器 （FRC）

bit 7-6 **FCKSM1:FCKSM0：**时钟切换和故障保护时钟监视器配置位
- 1x = 禁止时钟切换和故障保护时钟监视器
- 01 = 使能时钟切换，禁止故障保护时钟监视器
- 00 = 使能时钟切换和故障保护时钟监视器

bit 5 **OSCIOFCN：**OSCO 引脚配置位

如果 POSCMD1:POSCMD0 = 11 或 00：
- 1 = OSCO/CLKO/RC15 用作 CLKO （Fosc/2）
- 0 = OSCO/CLKO/RC15 用作端口 I/O （RC15）

如果 POSCMD1:POSCMD0 = 10 或 01：
OSCIOFCN 对 OSCO/CLKO/RC15 没有影响。

bit 4 **IOL1WAY：**IOLOCK 一次置 1 使能位
- 1 = IOLOCK 位（OSCCON<6>）可以置 1 一次，前提是已经完成解锁序列。一旦被置 1，就不能再次写入外设引脚选择寄存器。
- 0 = 完成解锁序列后，就可根据需要将 IOLOCK 位置 1 或清零。

bit 3 **未实现：**读为 1

bit 2 **I2C2SEL：**I2C2 引脚选择位 [(1)]
- 1 = 将 SCL2/SDA2 引脚用于 I2C2
- 0 = 将 ASCL2/ASDA2 引脚用于 I2C2

注 1： 仅在 100 引脚器件中实现；在其他器件中未实现，读为 1。

寄存器 24-2： CW2：闪存配置字 2 （续）

bit 1-0	**POSCMD1:POSCMD0：**主振荡器配置位 11 = 禁止主振荡器 10 = 选择 HS 振荡器模式 01 = 选择 XT 振荡器模式 00 = 选择 EC 振荡器模式

注 1： 仅在 100 引脚器件中实现；在其他器件中未实现，读为 1。

寄存器 24-3： CW3：闪存配置字 3

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	U-1	U-1	U-1	U-1	R/PO-1
WPEND	WPCFG	WPDIS	—	—	—	—	WPFP8
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
WPFP7	WPFP6	WPFP5	WPFP4	WPFP3	WPFP2	WPFP1	WPFP0
bit 7							bit 0

图注：			
R = 可读位	PO = 一次编程位	U = 未实现位，读为 0	
-n = 未对器件编程时的值		1 = 置 1	0 = 清零

bit 23-16	**未实现位：**读作 1
bit 15	**WPEND：**段写保护结束页选择位 1 = 受保护代码段的下边界位于程序存储器（000000h）的底端；上边界是由 WPFP8:WPFP0 指定的代码页 0 = 受保护代码段的上边界是程序存储器的最后一页；下边界是由 WPFP8:WPFP0 指定的代码页
bit 14	**WPCFG：**配置字代码页保护选择位 1 = 最后一页 （在程序存储器的顶部）和闪存配置字未受保护 0 = 最后一页和闪存配置字受代码保护
bit 13	**WPDIS：**段写保护禁止位 1 = 禁止段代码保护 0 = 使能段代码保护；受保护的段由 WPEND、 WPCFG 和 WPFPx 配置位定义
bit 12-9	**未实现：**读为 1
bit 8-0	**WPFP8:WPFP0：**受保护的代码段边界页位 指定受保护的代码段的边界为 16K 字程序代码页，从程序存储器底端的第 0 页开始。 如果 WPEND = 1： 指定代码页的结束地址为代码段的上边界。 如果 WPEND = 0： 指定代码页的起始地址是代码段的下边界。

寄存器 24-4： DEVID：器件 ID 寄存器

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23							bit 16

U	U	R	R	R	R	R	R
—	—	FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2
bit 15							bit 8

R	R	R	R	R	R	R	R
FAMID1	FAMID0	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
bit 7							bit 0

图注：	R = 只读位	U = 未实现位

bit 23-14 **未实现：**读为 `1`

bit 13-6 **FAMID7:FAMID0：**器件系列标识符位
`01000000` = PIC24FJ256GA110 系列

bit 5-0 **DEV5:DEV0：**各个器件的标识符位
`001000` = PIC24FJ128GA106
`001010` = PIC24FJ128GA108
`001110` = PIC24FJ128GA110
`010000` = PIC24FJ192GA106
`010010` = PIC24FJ192GA108
`010110` = PIC24FJ192GA110
`011000` = PIC24FJ256GA106
`011010` = PIC24FJ256GA108
`011110` = PIC24FJ256GA110

寄存器 24-5： DEVREV：器件版本寄存器

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23							bit 16

U	U	U	U	U	U	U	R
—	—	—	—	—	—	—	MAJRV2
bit 15							bit 8

R	R	U	U	U	R	R	R
MAJRV1	MAJRV0	—	—	—	DOT2	DOT1	DOT0
bit 7							bit 0

图注：	R = 只读位	U = 未实现位

bit 23-9 **未实现：**读为 `0`

bit 8-6 **MAJRV2:MAJRV0：**主要版本标识符位

bit 5-3 **未实现：**读为 `0`

bit 2-0 **DOT2:DOT0：**次要版本标识符位

24.2 片上稳压器

所有 PIC24FJ256GA110 系列器件使用标称的 2.5V 电压为其内核数字逻辑供电。对于需要工作在一个更高的典型电压值如 3.3V 的设计中，这可能会引起问题。为简化系统设计， PIC24FJ256GA110 系列中的所有器件均包含一个片内稳压器，可使器件内核逻辑在 VDD 下工作。

由 ENVREG 引脚控制稳压器。把 VDD 连到该引脚将使能稳压器，然后稳压器通过其他 VDD 引脚向内核供电。当使能稳压器时，必须在 VDDCORE/VCAP 引脚连接低 ESR 电容 （如陶瓷电容）（见图 24-1）。这可帮助维持稳压器的稳定性。**第 27.1 节 " 直流特性 "** 中提供了该滤波电容（CEFC）的推荐值。

如果 ENVREG 与 VSS 相连，则禁止稳压器。在这种情况下，必须通过 VDDCORE/VCAP 引脚对器件内核逻辑单独提供标称值为 2.5V 的电压，从而使 I/O 引脚可以具有较高的电压（通常为 3.3V）。另外，VDDCORE/VCAP 和 VDD 引脚可以连在一起，使器件工作在较低的标称电压下。参见图 24-1 了解可能的配置。

24.2.1 稳压器跟踪模式和低电压检测

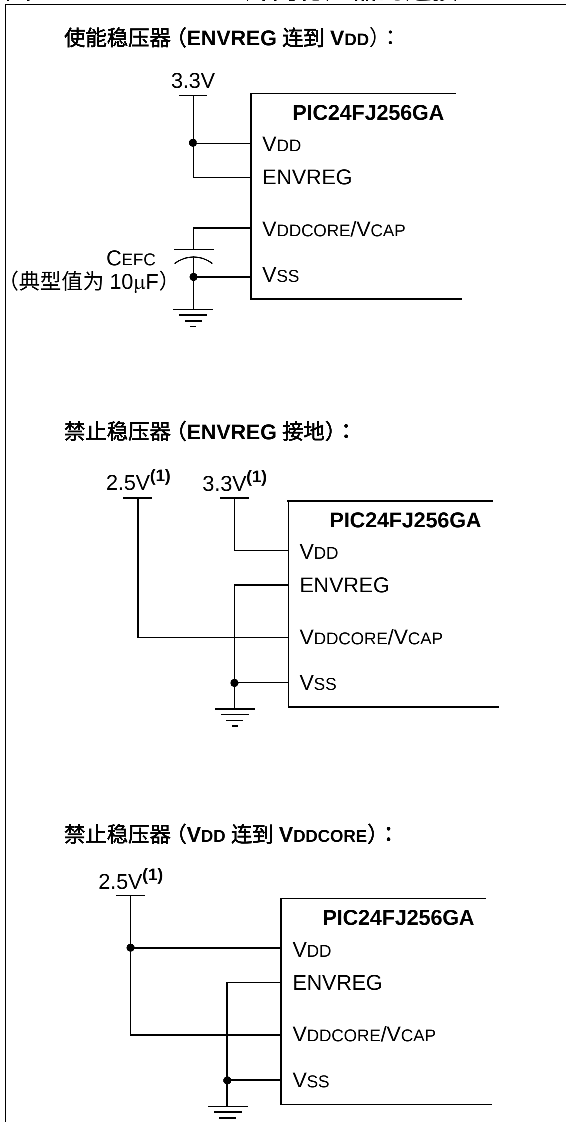
当使能时，片内稳压器为数字内核逻辑提供标称值为 2.5V 的恒定电压。

稳压器可提供从约为 2.5V 到器件所能承受的最高 VDDMAX 的 VDD 范围。该稳压器无法将低于 2.5V 的 VDD 电压提高。为了防止出现 " 欠压 " 的情况，当电压对于稳压器过低时，稳压器进入到跟踪模式。在跟踪模式下，稳压器输出跟随 VDD，通常低 100 mV。

器件进入跟踪模式后，不可能再继续以全速工作。为了提供有关器件何时进入跟踪模式的信息，片内稳压器中包含了一个简单的低电压检测电路。当 VDD 降低到全速工作电压以下时，该电路将低电压检测中断标志位 LVDIF （IFS4<8>）置 1。这可被用于产生中断并使应用进入到低功耗工作模式，或触发受控关断。

低电压检测仅在稳压器使能的情况下可用。

图 24-1： 片内稳压器的连接



注 1： 这些为典型的工作电压值。请参见**第 27.1 节 " 直流特性 "** 以了解 VDD 和 VDDCORE 的全工作电压范围。

24.2.2 片内稳压器和 POR

使能稳压器后，需等待大约 500 μs 才能产生输出。这段时间被称为 TSTARTUP，在这期间禁止代码执行。每次器件在掉电 （也包括休眠模式）之后恢复工作时都会产生 TSTARTUP 延时。

如果禁止稳压器，将自动使能上电延时定时器（PWRT）。在器件启动时， PWRT 会增加一段固定的 64 ms （标称值）的延时。

24.2.3 片内稳压器和 BOR

当使能片内稳压器时， PIC24FJ256GA110 系列器件也会有一个简单的欠压保护功能。如果向稳压器提供的电压不足以维持跟踪电压，那么稳压器复位电路将产生欠压复位。BOR 标志位（RCON<1>）捕捉该事件。在《PIC24FJ 系列参考手册》的**第 7 章 " 复位 "** （DS39712A_CN）中将提供欠压电压的规范。

24.2.4 上电要求

片内稳压器是为了满足器件的上电要求而设计的。如果应用不使用稳压器，那就必须严格遵守上电条件。在上电时， V_{DDCORE} 决不能超出 V_{DD} 电平 0.3V 以上。

> **注：** 更多信息，请参见**第 27.0 节 " 电气特性 "**。

24.2.5 稳压器待机模式

当使能时，片内稳压器总是消耗略大于 I_{DD}/I_{PD} 的电流，器件位于休眠模式时也是如此，尽管此时内核数字逻辑并不需要耗能。对于能源紧张的应用，为了节省更多的能源，只要器件进入休眠模式，稳压器就会被自动禁止。 VREGS （RCON<8>）位控制此功能。默认情况下，该位清零，使能待机模式。当从待机模式唤醒时，稳压器需要约 190 µs 的唤醒时间。这段额外的时间用于确保稳压器能提供足够的拉电流来为闪存供电。

对于需要较短唤醒时间的应用，可以禁止稳压器待机模式。 VREGS 位 （RCON<8>）可被置 1 以关闭待机模式，从而使闪存在休眠模式下一直处于上电状态，且器件可在 10 µs 内被唤醒。当 VREGS 置 1 时，休眠模式的功耗约比允许稳压器进入待机模式时高 40 µA。

24.3 看门狗定时器 （WDT）

对于 PIC24FJ256GA110 系列器件， WDT 由 LPRC 振荡器驱动。当使能 WDT 时，也将同时使能该时钟源。

由 LPRC 提供的 WDT 时钟源的频率标称值为 31 kHz。将此时钟源输入到可配置为 5 位 （32 分频）或 7 位（128 分频）工作模式的预分频器。具体工作模式由 FWPSA 配置位设置。31 kHz 的输入使预分频器生成标称的 WDT 超时周期（T_{WDT}）——5 位模式为 1 ms，7 位模式为 4 ms。

分频比可变的后分频器对 WDT 预分频器的输出进行分频并扩展超时周期范围。后分频比由 WDTPS3:WDTPS0 配置位 （CW1<3:0>）控制，该配置位共允许选择 16 种设置，从 1:1 到 1:32,768。使用预分频器和后分频器后，可获得 1 ms 到 131 秒的超时周期。

WDT、预分频器和后分频器在以下条件下复位：

- 在器件出现任何复位时
- 在时钟切换完成，无论切换是由软件 （即改变 NOSC 位后将 OSWEN 位置 1）或硬件 （即故障保护时钟监视器）引起
- 当执行 `PWRSAV` 指令时 （即进入休眠模式或空闲模式）
- 当器件退出休眠模式或空闲模式恢复正常工作时
- 当在正常执行过程中使用 `CLRWDT` 指令时

如果使能 WDT，它将在休眠或空闲模式下继续运行。当发生 WDT 超时，将唤醒器件并且代码将继续从 `PWRSAV` 指令处开始执行。器件被唤醒后，需要用软件将相应的 SLEEP 或 IDLE 位 （RCON<3:2>）清零。

WDT 标志位 WDTO（RCON<4>）不会在 WDT 超时后自动清零。要检测后续的 WDT 事件，必须用软件将该标志清零。

> **注：** 当执行 `CLRWDT` 和 `PWRSAV` 指令时，预分频器和后分频器的计数值将被清零。

24.3.1 窗口操作

看门狗定时器有一个可选的固定窗口工作模式。在窗口模式下， CLRWDT 指令仅能在 WDT 编程周期的后 1/4 时间段内复位 WDT。在该窗口前执行的 CLRWDT 指令会导致 WDT 复位，类似于 WDT 超时。

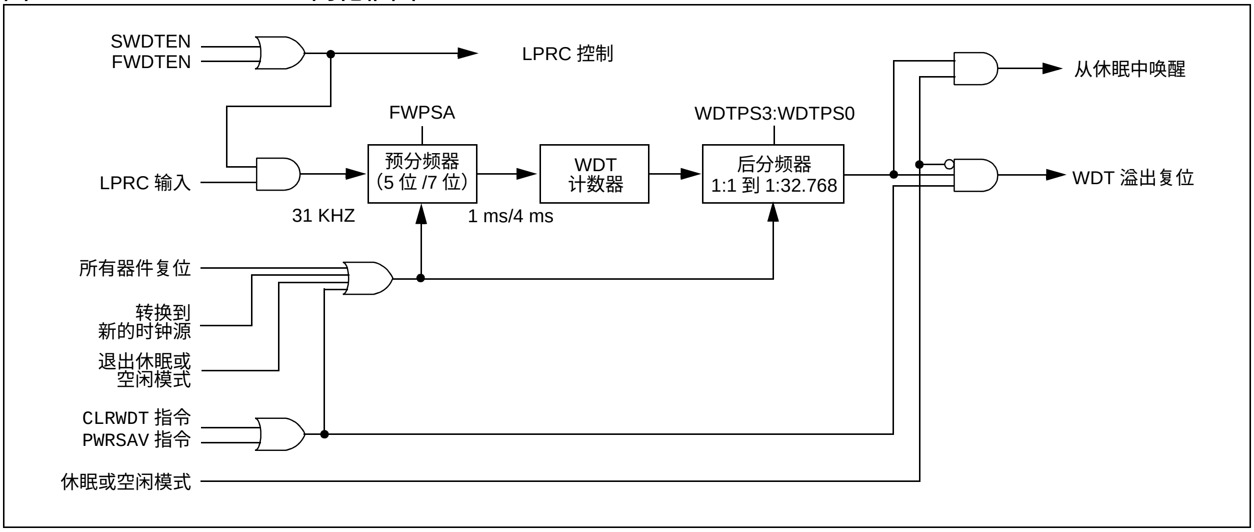
通过将 WINDIS 配置位（CW1<6>）编程为 0 以使能窗口 WDT 模式。

24.3.2 控制寄存器

用FWDTEN配置位来使能或禁止WDT。FWDTEN配置位置 1 时， WDT 使能。

FWDTEN 配置位被编程为 0 后，可使用软件有选择地控制 WDT。通过将 SWDTEN 控制位（RCON<5>）置1来使能WDT。任何类型的器件复位都会使SWDTEN控制位清零。软件控制 WDT 选项允许用户在关键代码段使能 WDT 并在非关键代码段禁止 WDT，以最大限度地降低功耗。

图 24-2： WDT 简化框图



24.4 程序校验和代码保护

PIC24FJ256GA110 系列器件提供了两种补充方法来防止应用程序代码被改写和擦除。同样可以防止在运行过程中器件配置被意外更改。

24.4.1 通用段保护

对于 PIC24FJ256GA110 系列中的所有器件，片内程序存储空间被当作一个单独的块，称为通用段 （General Segment，GS）。配置位 GCP 控制该块的代码保护。该位阻止外部对程序存储空间的读写操作。但对正常的执行模式没有直接影响。

写保护是由配置字中的 GWRP 位控制的。当 GWRP 被编程为 0 时，禁止在内部写和擦除程序存储器。

24.4.2 代码段保护

除了全局通用段保护外，可以单独保护程序存储空间的独立子区域以防止对其进行写和擦除操作。该区域对于一些单独需要防止进行写和擦除操作的代码保护块有很多用途，例如自举应用程序。不同于常见的引导区，用户可在程序空间内任何位置定位 PIC24FJ256GA110 系列器件中的任意大小的特定受保护段。

代码段保护通过在写或擦除地址位于指定范围内时禁止NVM安全互锁，以此向程序存储器的指定区域提供了新增的保护等级。代码段保护不会改写由 GCP 或 GWRP 位所控制的通用段保护。例如，如果使能 GCP 和 GWRP，那么使能程序存储器的低半部分的段代码保护不会取消高半部分的通用段保护。

受保护代码段的大小和类型由配置字 3 中的 WPFPx、WPEND、WPCFG 和 WPDIS 位来进行配置。通过对 WPDIS 位进行编程 （= 0）以使能代码段保护。WPFP 位通过指定受保护段的起始或结束地址的 512 字的代码页确定被保护段的大小。指定区域也被包含在内，因此页也被保护。

WPEND 位确定被保护段是否将程序空间的顶部和底部作为边界。对 WPEND 进行编程 （= 0）可以把程序存储器的底部 （000000h）设置为被保护段的下边界。不对 WPEND 位进行编程 （= 1）可以对指定页到已实现程序存储器的最末页进行保护，包括配置字单元。

独立位 WPCFG 用于单独保护程序空间的最后一页，包括闪存配置字。对 WPCFG 进行编程 （= 0）可以保护最后一页，与其他位的设置无关。在需要对存储器底部的代码段和闪存配置字同时进行写保护的情况下，对 WPCFG 进行编程是有用的。

段代码保护的各种选项如表 24-2 所示。

24.4.3 配置寄存器保护

有两种方法可以保护配置寄存器，使其免遭意外改写或读取。主要的保护方式与保护 RP 寄存器的方法相同——影子寄存器中包含了一个基准值，持续将该值与实际的值进行比较。

为了防范不可预见事件造成损害，由于单独的存储单元故障 （如 ESD 事件）引起的配置位更改将导致奇偶校验错误并触发器件复位。

配置寄存器的数据来自于程序存储器中的闪存配置字。当 GCP 位置 1 时，也将保护器件配置的源数据。即使未使能通用段保护，也可以通过使用适当的代码段保护设置来保护器件配置。

表 24-2： 段代码保护配置选项

段配置位			代码段的写 / 擦除保护
WPDIS	WPEND	WPCFG	
1	X	1	没有使能其他保护；所有程序存储器保护由 GCP 和 GWRP 配置
1	X	0	最后一个代码页受保护，包括闪存配置字
0	1	0	保护从 WPFP8:WPFP0 定义的代码页的首地址，到实现的受保护程序存储器（包括在内）的尾地址，包括闪存配置字
0	0	0	保护从 000000h 到受保护的由 WPFP8:WPFP0 （包括在内）定义的代码页的尾地址
0	1	1	保护从由 WPFP8:WPFP0 定义的代码页的首地址，到实现的受保护程序存储器（包括在内）的尾地址，包括闪存配置字。
0	0	1	保护从 WPFP8:WPFP0 定义的代码页的首地址，到实现的受保护程序存储器（包括在内）的尾地址。

 超前信息

24.5 JTAG 接口

PIC24FJ256GA110 系列器件实现了 JTAG 接口，以支持边界扫描器件测试和在线串行编程。

24.6 在线串行编程

PIC24FJ256GA110 系列单片机可以在最终应用电路中进行串行编程。只需要五根线即可完成这一操作，其中时钟线（PGECx）和数据线（PGEDx）各一根，其余三根分别是电源线、接地线和编程电压线。这允许用户使用未编程器件生产电路板，而仅在产品交付前才对单片机进行编程，从而可将最新版本的固定或定制固件烧写到器件中。

24.7 在线调试器

选择MPLAB® ICD2作为调试器时，使能在线调试功能。这一功能允许结合 MPLAB IDE 进行一些简单的调试。通过 PGECx（仿真 / 调试时钟）和 PGEDx（仿真 / 调试数据）引脚功能控制调试功能。

要使用器件的在线调试功能，在设计中必须实现 MCLR、VDD、VSS 以及由 ICS 配置位指定的 PGECx/PGEDx 引脚对的 ICSP 连接。此外，当使能该功能时，某些资源就不能用于一般用途了。这些资源包括数据 RAM 的前 80 个字节和两个 I/O 引脚。

注：

25.0 开发支持

一系列硬件及软件开发工具对 PIC® 单片机提供支持：

- 集成开发环境
  - MPLAB® IDE 软件
- 汇编器 / 编译器 / 链接器
  - MPASM™ 汇编器
  - MPLAB C18 和 MPLAB C30 C 编译器
  - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
  - MPLAB ASM30 汇编器 / 链接器 / 库
- 模拟器
  - MPLAB SIM 软件模拟器
- 仿真器
  - MPLAB ICE 2000 在线仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器
  - MPLAB ICD 2
- 器件编程器
  - PICSTART® Plus 开发编程器
  - MPLAB PM3 器件编程器
  - PICkit™ 2 开发编程器
- 低成本演示和开发板及评估工具包

25.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16 位单片机市场提供了前所未有的易于使用的软件开发平台。MPLAB IDE 是基于 Windows® 操作系统的应用软件，包括：

- 一个包含所有调试工具的图形界面
  - 模拟器
  - 编程器 （单独销售）
  - 仿真器 （单独销售）
  - 在线调试器 （单独销售）
- 具有彩色上下文代码显示的全功能编辑器
- 多项目管理器
- 内容可直接编辑的可定制式数据窗口
- 高级源代码调试
- 可视化器件初始化程序，便于进行寄存器的初始化
- 鼠标停留在变量上进行查看的功能
- 通过拖放把变量从源代码窗口拉到观察窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 HI-TECH 软件 C 编译器和 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件 （汇编语言或 C 语言）
- 点击一次即可完成汇编 （或编译）并将代码下载到 PIC MCU 仿真器和模拟器工具中 （自动更新所有项目信息）
- 可使用如下各项进行调试：
  - 源文件 （汇编语言或 C 语言）
  - 混合汇编语言和 C 语言
  - 机器码

MPLAB IDE 在单个开发范例中支持使用多种调试工具，包括从成本效益高的模拟器到低成本的在线调试器，再到全功能的仿真器。这样缩短了用户升级到更加灵活而功能更强大的工具时的学习时间。

25.2 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于所有的 PIC MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特征：

- 集成在 MPLAB IDE 项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

25.3 MPLAB C18 和 MPLAB C30 C 编译器

MPLAB C18 和 MPLAB C30 代码开发系统是完全的 ANSI C 编译器，分别适用于 Microchip 的 PIC18 和 PIC24 系列单片机及 dsPIC30F 和 dsPIC33 系列数字信号控制器。这些编译器可提供其他编译器并不具备的强大的集成功能和出众的代码优化能力，且使用方便。

为便于源代码调试，编译器提供了针对 MPLAB IDE 调试器的优化符号信息。

25.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

MPLIB目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用中。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特征：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起来增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

25.5 MPLAB ASM30 汇编器、链接器和库管理器

MPLAB ASM30 汇编器为 dsPIC30F 器件提供转换自符号汇编语言的可重定位机器码。 MPLAB C30 C 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特征：

- 支持整个 dsPIC30F 指令集
- 支持定点数据和浮点数据
- 命令行界面
- 丰富的指令集
- 灵活的宏语言
- MPLAB IDE 兼容性

25.6 MPLAB SIM 软件模拟器

MPLAB SIM软件模拟器在指令级对PIC MCU和dsPIC® DSC 进行模拟，使得用户可以在 PC 主机的环境下进行代码开发。对于任何给定的指令，用户均可对数据区进行检查或修改，并通过各种触发机制来产生激励。可以将各寄存器的情况记录在文件中，以便进行进一步地运行时分析。跟踪缓冲器和逻辑分析器的显示使模拟器还能记录和跟踪程序的执行、 I/O 的动作、大部分的外设及内部寄存器的状况。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C18 和 MPLAB C30 C 编译器以及MPASM和MAPLAB ASM30 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

25.7 MPLAB ICE 2000 高性能在线仿真器

MPLAB ICE 2000 在线仿真器旨在为产品开发工程师提供一整套用于 PIC 单片机的设计工具。 MPLAB ICE 2000 在线仿真器的软件控制由 MPLAB 集成开发环境平台提供，它允许在单一环境下进行编辑、编译、下载以及源代码调试。

MPLAB ICE 2000 是全功能仿真器系统，它具有增强的跟踪、触发和数据监控功能。处理器模块可插拔，使系统可轻松进行重新配置以适应各种不同处理器的仿真需要。MPLAB ICE 2000 在线仿真器的架构允许对其进行扩展以支持新的 PIC 单片机。

MPLAB ICE 2000 在线仿真器系统设计为一款实时仿真系统，该仿真系统具备通常只有昂贵的开发工具中才有的高级功能。选择 PC 平台和 Microsoft® Windows® 32 位操作系统可使这些功能在一个简单而统一的应用中得到很好的利用。

25.8 MPLAB REAL ICE 在线仿真器系统

MPLAB REAL ICE在线仿真器系统是Microchip针对其闪存 DSC 和 MCU 器件而推出的新一代高速仿真器。结合 MPLAB 集成开发环境（IDE）所具有的易于使用且功能强大的图形用户界面，该仿真器可对 PIC® 闪存 MCU 和 dsPIC® DSC 进行调试和编程。IDE 是随每个工具包一起提供的。

MPLAB REAL ICE 探针通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与常用 MPLAB ICD 2 系统兼容的连接器 （RJ11）或新型抗噪声、高速低压差分信号（LVDS）互连电缆 （CAT5）与目标板相连。

可通过 MPLAB IDE 下载将来版本的固件，对 MPLAB REAL ICE 进行现场升级。在即将推出的 MPLAB IDE 版本中，会支持许多新器件，还将增加一些新特性，如软件断点和汇编代码跟踪等。在同类仿真器中， MPLAB REAL ICE 的优势十分明显：低成本、高速仿真、实时变量监视、跟踪分析、复杂断点、耐用的探针接口及较长（长达 3 米）的互连电缆。

25.9 MPLAB ICD 2 在线调试器

Microchip 的在线调试器 MPLAB ICD 2 是一款功能强大而成本低廉的运行时开发工具，通过 RS-232 或高速 USB 接口与 PC 主机相连。该工具基于闪存 PIC MCU，可用于开发本系列及其他 PIC MCU 和 dsPIC DSC。MPLAB ICD 2 使用了闪存器件中内建的在线调试功能。该功能结合 Microchip 的在线串行编程（In-Circuit Serial Programming™， ICSP™）协议，可在 MPLAB 集成开发环境的图形用户界面上提供成本效益很高的在线闪存调试。这使设计人员可通过设置断点、单步运行以及对变量、CPU 状态以及外设寄存器进行监视的方法实现源代码的开发和调试。其全速运行特性可对硬件和应用进行实时测试。 MPLAB ICD 2 还可用作某些 PIC 器件的开发编程器。

25.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款通用的、符合 CE 规范的器件编程器，其可编程电压设置在 V_{DDMIN} 和 V_{DDMAX} 之间时可靠性最高。它有一个用来显示菜单和错误信息的大 LCD 显示器（128 x 64），以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP™ 电缆。在单机模式下， MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC 器件进行读取、验证和编程。在该模式下它还可设置代码保护。 MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对存储器很大的器件进行快速编程，它还采用 SD/MMC 卡用作文件存储及数据安全应用。

25.11 PICSTART Plus 开发编程器

PICSTART Plus开发编程器是一款易于使用而成本低廉的原型编程器。它通过 COM （RS-232）端口与 PC 相连。 MPLAB 集成开发环境软件使得该编程器的使用简便、高效。PICSTART Plus 开发编程器支持采用 DIP 封装的大部分 PIC 器件，其引脚数最多可达 40 个。引脚数更多的器件，如 PIC16C92X 和 PIC17C76X，可通过连接一个转接插槽来获得支持。PICSTART Plus 开发编程器符合 CE 规范。

25.12 PICkit 2 开发编程器

PICkit™ 2 开发编程器是一个低成本编程器；对于某些选定闪存器件，它也是一个调试器，通过其易于使用的接口可对众多 Microchip 的低档、中档和 PIC18F 系列闪存单片机进行编程。 PICkit 2 入门工具包中包含一个有实验布线区的开发板、十二堂系列课程、软件和 HI-TECH 的 PICC™ Lite C 编译器，有助于用户快速掌握 PIC® 单片机的使用。这一工具包为使用 Microchip 功能强大的中档闪存系列单片机进行编程、评估和应用开发，提供了所需的一切。

25.13 演示、开发和评估板

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于测试和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、 RS-232 接口、 LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外， Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、 KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart 电池管理、SEEVAL® 评估系统、 Σ–Δ ADC、流速传感器，等等。

有关演示、开发和评估工具包的完整列表，请查阅 Microchip 公司网页 （www.microchip.com）。

26.0 指令集综述

注：	本章是 PIC24F 指令集架构的简要概述，并不能作为详尽的参考资料使用。

PIC24F 指令集与以前的 PIC® MCU 指令集相比，添加了许多增强功能，并保持了易于从其他 PIC MCU 指令集移植的特点。大部分指令只占用一个程序存储字。只有 3 条指令需要两个程序存储单元。

每条单字指令都是一个 24 位字，由一个 8 位的操作码（指明指令类型）和一个或多个操作数 （指定指令具体操作）组成。整个指令集具有高度的正交性，分为以下 4 种基本类型：

- 面向字或字节的操作类指令
- 面向位的操作类指令
- 立即数操作类指令
- 控制操作类指令

表26-1给出了说明指令时所要用到的通用符号。表26-2 中的 PIC24F 指令集汇总列出了所有指令及每条指令影响的状态位。

大部分面向字或字节的 W 寄存器指令 （包含桶形移位寄存器指令）含有三个操作数：

- 第一个源操作数通常是不带地址修改量的 Wb 寄存器
- 第二个源操作数通常是带或不带地址修改量的 Ws 寄存器
- 结果的目标地址通常是带或不带地址修改量的 Wd 寄存器。

而面向字或字节的文件寄存器指令具有两个操作数：

- 文件寄存器 （由 f 指定）
- 目标寄存器 （可以是文件寄存器 f 也可以是记作 WREG 的 W0 寄存器）

大部分面向位的操作类指令（包括简单循环/移位指令）含有两个操作数：

- W 寄存器 （带或不带地址修改量）或文件寄存器 （由 Ws 或 f 的值指定）
- W 寄存器或文件寄存器中的位 （由立即数直接指定或由 Wb 寄存器中的内容间接指定）

涉及数据传送的立即数指令可能使用以下操作数：

- 将被装载到 W 寄存器或文件寄存器的立即数值 （由 k 指定）
- 将要装载立即数的 W 寄存器或文件寄存器 （由 Wb 或 f 指定）

而涉及算术或逻辑运算的立即数指令使用以下操作数：

- 第一个源操作数是不带地址修改量的 Wb 寄存器
- 第二个源操作数为立即数
- 结果的目标地址 （只有在与第一个源操作数不同的情况下）通常为带或不带地址修改量的 Wd 寄存器

控制操作类指令使用以下操作数：

- 程序存储器地址
- 表读和表写指令的模式

除某些双字指令外所有指令都是单字指令。双字指令中所有必需的信息都在这 48 位中，第二个字的高 8 位全为 `0`。如果第二个字作为一条指令执行，它将会被当作 `NOP` 指令执行。

除非条件测试结果为真或者执行后改变了程序计数器的值，否则执行所有的单字指令都只需要一个指令周期。对于上述两种特殊情况，指令执行需要两个指令周期，第二个指令周期执行一条 `NOP` 指令。值得注意的特殊指令有：`BRA`（无条件/计算转移指令）、间接 `CALL/GOTO` 指令、所有表读和表写指令以及 `RETURN/RETFIE` 指令，这些指令都是单字指令，但执行起来需要两或三个指令周期。

某些涉及跳过下一条指令的指令，在执行跳过时需要两或三个指令周期，具体周期数取决于被跳过的指令是单字指令还是双字指令。此外需要传送两个字的指令需要两个周期。执行双字指令需要两个指令周期。

表 26-1：　操作码说明中使用的符号

字段	说明
#text	表示由 `text` 定义的立即数
(text)	表示 `text` 的内容
[text]	表示地址为 `text` 的单元
{ }	可选字段或操作
<n:m>	寄存器位域
.b	字节模式选择
.d	双字模式选择
.S	影子寄存器选择
.w	字模式选择 （默认情况）
bit4	4 位位选择字段 （用于字寻址指令） ∈ {0...15}
C、 DC、 N、 OV 和 Z	MCU 状态位：进位标志位、半进位标志位、负标志位、溢出标志位和全零标志位
Expr	绝对地址、标号或表达式 （由链接器解析）
f	文件寄存器地址∈ {0000h...1FFFh}
lit1	1 位无符号立即数∈ {0,1}
lit4	4 位无符号立即数∈ {0...15}
lit5	5 位无符号立即数∈ {0...31}
lit8	8 位无符号立即数∈ {0...255}
lit10	10 位无符号立即数，字节模式下∈ {0...255}，字模式下∈ {0:1023}
lit14	14 位无符号立即数∈ {0...16384}
lit16	16 位无符号立即数∈ {0...65535}
lit23	23 位无符号立即数∈ {0...8388608} ； LSB 必须为 `0`
None	该字段不必有输入项，可以为空白
PC	程序计数器
Slit10	10 位有符号立即数∈ {-512...511}
Slit16	16 位有符号立即数∈ {-32768...32767}
Slit6	6 位有符号立即数∈ {-16...16}
Wb	基本 W 寄存器∈ {W0..W15}
Wd	目标 W 寄存器∈ {Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	目标 W 寄存器∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	被除数和除数工作寄存器对 （直接寻址）
Wn	16 个工作寄存器之一∈ {W0..W15}
Wnd	16 个目标工作寄存器之一∈ {W0..W15}
Wns	16 个源工作寄存器之一∈ {W0..W15}
WREG	W0 （文件寄存器指令中使用的工作寄存器）
Ws	源 W 寄存器∈ {Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	源 W 寄存器∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }

表 26-2： 指令集汇总

汇编指令助记符	汇编语法		说明	字数	周期数	受影响的状态标志位
ADD	ADD	f	f = f + WREG	1	1	C、DC、N、OV和Z
	ADD	f,WREG	WREG = f + WREG	1	1	C、DC、N、OV和Z
	ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C、DC、N、OV和Z
	ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C、DC、N、OV和Z
	ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C、DC、N、OV和Z
ADDC	ADDC	f	f = f + WREG + (C)	1	1	C、DC、N、OV和Z
	ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C、DC、N、OV和Z
	ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C、DC、N、OV和Z
	ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C、DC、N、OV和Z
	ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C、DC、N、OV和Z
AND	AND	f	f = f .AND. WREG	1	1	N 和 Z
	AND	f,WREG	WREG = f .AND. WREG	1	1	N 和 Z
	AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N 和 Z
	AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N 和 Z
	AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N 和 Z
ASR	ASR	f	f = 算术右移 f	1	1	C、 N、 OV 和 Z
	ASR	f,WREG	WREG = 算术右移 f	1	1	C、 N、 OV 和 Z
	ASR	Ws,Wd	Wd = 算术右移 Ws	1	1	C、 N、 OV 和 Z
	ASR	Wb,Wns,Wnd	Wnd = 将 Wb 算术右移 Wns 位	1	1	N 和 Z
	ASR	Wb,#lit5,Wnd	Wnd = 将 Wb 算术右移 lit5 位	1	1	N 和 Z
BCLR	BCLR	f,#bit4	将 f 中的指定位清零	1	1	无
	BCLR	Ws,#bit4	将 Ws 中的指定位清零	1	1	无
BRA	BRA	C,Expr	如果进位位为 1 则转移	1	1 (2)	无
	BRA	GE,Expr	如果大于或等于则转移	1	1 (2)	无
	BRA	GEU,Expr	如果无符号大于或等于则转移	1	1 (2)	无
	BRA	GT,Expr	如果大于则转移	1	1 (2)	无
	BRA	GTU,Expr	如果无符号大于则转移	1	1 (2)	无
	BRA	LE,Expr	如果小于或等于则转移	1	1 (2)	无
	BRA	LEU,Expr	如果无符号小于或等于则转移	1	1 (2)	无
	BRA	LT,Expr	如果小于则转移	1	1 (2)	无
	BRA	LTU,Expr	如果无符号小于则转移	1	1 (2)	无
	BRA	N,Expr	如果为负则转移	1	1 (2)	无
	BRA	NC,Expr	如果进位位为 0 则转移	1	1 (2)	无
	BRA	NN,Expr	如果非负则转移	1	1 (2)	无
	BRA	NOV,Expr	如果未溢出则转移	1	1 (2)	无
	BRA	NZ,Expr	如果非零则转移	1	1 (2)	无
	BRA	OV,Expr	如果溢出则转移	1	1 (2)	无
	BRA	Expr	无条件转移	1	2	无
	BRA	Z,Expr	如果为零则转移	1	1 (2)	无
	BRA	Wn	计算转移	1	2	无
BSET	BSET	f,#bit4	将 f 中的指定位置 1	1	1	无
	BSET	Ws,#bit4	将 Ws 中的指定位置 1	1	1	无
BSW	BSW.C	Ws,Wb	将 C 位内容写入 Ws<Wb>	1	1	无
	BSW.Z	Ws,Wb	将 Z 位内容写入 Ws<Wb>	1	1	无
BTG	BTG	f,#bit4	将 f 中的某位取反	1	1	无
	BTG	Ws,#bit4	将 Ws 中的某位取反	1	1	无
BTSC	BTSC	f,#bit4	对 f 中的指定位进行测试，如果为零则跳过	1	1 (2 或 3)	无
	BTSC	Ws,#bit4	对 Ws 中的指定位进行测试，如果为零则跳过	1	1 (2 或 3)	无

表 26-2： 指令集汇总 （续）

汇编指令助记符	汇编语法	说明	字数	周期数	受影响的状态标志位
BTSS	BTSS f,#bit4	对 f 中的指定位进行测试，如果为 1 则跳过	1	1 （2 或 3）	无
	BTSS Ws,#bit4	对 Ws 中的指定位进行测试，如果为 1 则跳过	1	1 （2 或 3）	无
BTST	BTST f,#bit4	对 f 中的指定位进行测试	1	1	Z
	BTST.C Ws,#bit4	对 Ws 中的指定位进行测试，并将被测试位的值存储到进位标志位 C 中	1	1	C
	BTST.Z Ws,#bit4	对 Ws 中的指定位进行测试，并将被测试位的反码存储到全零标志位 Z 中	1	1	Z
	BTST.C Ws,Wb	对 Ws<Wb> 位进行测试，并将被测试位的值存储到进位标志位 C 中	1	1	C
	BTST.Z Ws,Wb	对 Ws<Wb> 位进行测试，并将被测试位的反码存储到全零标志位 Z 中	1	1	Z
BTSTS	BTSTS f,#bit4	对 f 中的指定位进行测试，并将 f 中的该位置 1	1	1	Z
	BTSTS.C Ws,#bit4	对 Ws 中的指定位进行测试，并将被测试位的值存储到进位标志位 C 中，然后将 Ws 中的该位置 1	1	1	C
	BTSTS.Z Ws,#bit4	对 Ws 中的指定位进行测试，并将被测试位的反码存储到全零标志位 Z 中，然后将 Ws 中的该位置 1	1	1	Z
CALL	CALL lit23	调用子程序	2	2	无
	CALL Wn	间接调用子程序	1	2	无
CLR	CLR f	f = 0x0000	1	1	无
	CLR WREG	WREG = 0x0000	1	1	无
	CLR Ws	Ws = 0x0000	1	1	无
CLRWDT	CLRWDT	将看门狗定时器清零	1	1	WDTO 和 Sleep
COM	COM f	f = $\overline{f}$	1	1	N 和 Z
	COM f,WREG	WREG = $\overline{f}$	1	1	N 和 Z
	COM Ws,Wd	Wd = $\overline{Ws}$	1	1	N 和 Z
CP	CP f	比较 f 和 WREG	1	1	C、DC、N、OV 和 Z
	CP Wb,#lit5	比较 Wb 和 lit5	1	1	C、DC、N、OV 和 Z
	CP Wb,Ws	比较 Wb 和 Ws (Wb - Ws)	1	1	C、DC、N、OV 和 Z
CP0	CP0 f	比较 f 和 0x0000	1	1	C、DC、N、OV 和 Z
	CP0 Ws	比较 Ws 和 0x0000	1	1	C、DC、N、OV 和 Z
CPB	CPB f	比较 f 和 WREG （通过带借位减法实现）	1	1	C、DC、N、OV 和 Z
	CPB Wb,#lit5	比较 Wb 和 lit5 （通过带借位减法实现）	1	1	C、DC、N、OV 和 Z
	CPB Wb,Ws	比较 Wb 和 Ws （通过带借位减法实现） （Wb - Ws - $\overline{C}$）	1	1	C、DC、N、OV 和 Z
CPSEQ	CPSEQ Wb,Wn	比较 Wb 和 Wn，如果相等则跳过	1	1 （2 或 3）	无
CPSGT	CPSGT Wb,Wn	比较 Wb 和 Wn，如果大于则跳过	1	1 （2 或 3）	无
CPSLT	CPSLT Wb,Wn	比较 Wb 和 Wn，如果小于则跳过	1	1 （2 或 3）	无
CPSNE	CPSNE Wb,Wn	比较 Wb 和 Wn，如果不相等则跳过	1	1 （2 或 3）	无
DAW	DAW Wn	Wn = 对 Wn 进行十进制调整	1	1	C
DEC	DEC f	f = f - 1	1	1	C、DC、N、OV 和 Z
	DEC f,WREG	WREG = f - 1	1	1	C、DC、N、OV 和 Z
	DEC Ws,Wd	Wd = Ws - 1	1	1	C、DC、N、OV 和 Z
DEC2	DEC2 f	f = f - 2	1	1	C、DC、N、OV 和 Z
	DEC2 f,WREG	WREG = f - 2	1	1	C、DC、N、OV 和 Z
	DEC2 Ws,Wd	Wd = Ws - 2	1	1	C、DC、N、OV 和 Z
DISI	DISI #lit14	在 k 个指令周期内禁止中断	1	1	无
DIV	DIV.SW Wm,Wn	有符号 16/16 位整数除法	1	18	N、Z、C 和 OV
	DIV.SD Wm,Wn	有符号 32/16 位整数除法	1	18	N、Z、C 和 OV
	DIV.UW Wm,Wn	无符号 16/16 位整数除法	1	18	N、Z、C 和 OV
	DIV.UD Wm,Wn	无符号 32/16 位整数除法	1	18	N、Z、C 和 OV
EXCH	EXCH Wns,Wnd	将 Wns 和 Wnd 交换	1	1	无
FF1L	FF1L Ws,Wnd	从左边 （MSb）查找第一个 1	1	1	C
FF1R	FF1R Ws,Wnd	从右边 （MSb）查找第一个 1	1	1	C

超前信息

表 26-2： 指令集汇总 （续）

汇编指令助记符	汇编语法	说明	字数	周期数	受影响的状态标志位
GOTO	GOTO Expr	转移到地址	2	2	无
	GOTO Wn	间接转移到地址	1	2	无
INC	INC f	f = f + 1	1	1	C、DC、N、OV和Z
	INC f,WREG	WREG = f + 1	1	1	C、DC、N、OV和Z
	INC Ws,Wd	Wd = Ws + 1	1	1	C、DC、N、OV和Z
INC2	INC2 f	f = f +2	1	1	C、DC、N、OV和Z
	INC2 f,WREG	WREG = f +2	1	1	C、DC、N、OV和Z
	INC2 Ws,Wd	Wd = Ws +2	1	1	C、DC、N、OV和Z
IOR	IOR f	f = f .IOR. WREG	1	1	N 和 Z
	IOR f,WREG	WREG = f .IOR. WREG	1	1	N 和 Z
	IOR #lit10,Wn	Wd = lit10 .IOR. Wd	1	1	N 和 Z
	IOR Wb,Ws,Wd	Wd = Wb .IOR. Ws	1	1	N 和 Z
	IOR Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N 和 Z
LNK	LNK #lit14	分配堆栈帧	1	1	无
LSR	LSR f	f = 逻辑右移 f	1	1	C、 N、 OV 和 Z
	LSR f,WREG	WREG = 逻辑右移 f	1	1	C、 N、 OV 和 Z
	LSR Ws,Wd	Wd = 逻辑右移 Ws	1	1	C、 N、 OV 和 Z
	LSR Wb,Wns,Wnd	Wnd = 将 Wb 逻辑右移 Wns 位	1	1	N 和 Z
	LSR Wb,#lit5,Wnd	Wnd = 将 Wb 逻辑右移 lit5 位	1	1	N 和 Z
MOV	MOV f,Wn	将 f 中的内容送入 Wn	1	1	无
	MOV [Wns+Slit10],Wnd	将 [Wns+Slit10] 中的内容送入 Wnd	1	1	无
	MOV f	将 f 中的内容送入目的寄存器	1	1	N 和 Z
	MOV f,WREG	将 f 中的内容送入 WREG	1	1	N 和 Z
	MOV #lit16,Wn	将 16 位立即数送入 Wn	1	1	无
	MOV.b #lit8,Wn	将 8 位立即数送入 Wn	1	1	无
	MOV Wn,f	将 Wn 中的内容送入 f	1	1	无
	MOV Wns,[Wns+Slit10]	将 Wns 中的内容送入 [Wns+Slit10]	1	1	
	MOV Wso,Wdo	将 Ws 中的内容送入 Wd	1	1	无
	MOV WREG,f	将 WREG 中的内容送入 f	1	1	N 和 Z
	MOV.D Wns,Wd	将 W(ns):W(ns + 1) 中的双字内容送入 Wd	1	2	无
	MOV.D Ws,Wnd	将 Ws 中的双字内容送入 W(nd + 1):W(nd)	1	2	无
MUL	MUL.SS Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Signed(Ws)	1	1	无
	MUL.SU Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	无
	MUL.US Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	无
	MUL.UU Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	无
	MUL.SU Wb,#lit5,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	无
	MUL.UU Wb,#lit5,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	无
	MUL f	W3:W2 = f * WREG	1	1	无
NEG	NEG f	f = $\overline{f}$ + 1	1	1	C、DC、N、OV和Z
	NEG f,WREG	WREG = $\overline{f}$ + 1	1	1	C、DC、N、OV和Z
	NEG Ws,Wd	Wd = $\overline{Ws}$ + 1	1	1	C、DC、N、OV和Z
NOP	NOP	空操作	1	1	无
	NOPR	空操作	1	1	无
POP	POP f	将栈顶 （TOS）的内容弹出到 f	1	1	无
	POP Wdo	将栈顶 （TOS）的内容弹出到 Wdo	1	1	无
	POP.D Wnd	从栈顶 （TOS）弹出双字到 W(nd):W(nd+1)	1	2	无
	POP.S	将影子寄存器的内容弹出到主寄存器	1	1	全部
PUSH	PUSH f	将 f 的内容压入栈顶 （TOS）	1	1	无
	PUSH Wso	将 Wso 的内容压入栈顶 （TOS）	1	1	无
	PUSH.D Wns	将 W(ns):W(ns + 1) 中的双字内容压入栈顶 （TOS）	1	2	无
	PUSH.S	将主寄存器中的双字内容压入影子寄存器	1	1	无

表 26-2： 指令集汇总 （续）

汇编指令助记符	汇编语法		说明	字数	周期数	受影响的状态标志位
PWRSAV	PWRSAV	#lit1	进入休眠或空闲模式	1	1	WDTO 和 Sleep
RCALL	RCALL	Expr	相对调用	1	2	无
	RCALL	Wn	计算调用	1	2	无
REPEAT	REPEAT	#lit14	将下一条指令重复执行 lit14 + 1 次	1	1	无
	REPEAT	Wn	将下一条指令重复执行 (Wn) + 1 次	1	1	无
RESET	RESET		软件器件复位	1	1	无
RETFIE	RETFIE		从中断返回	1	3 (2)	无
RETLW	RETLW	#lit10,Wn	返回并将立即数存入 Wn	1	3 (2)	无
RETURN	RETURN		从子程序返回	1	3 (2)	无
RLC	RLC	f	f = 对 f 执行带进位的循环左移	1	1	C、 N 和 Z
	RLC	f,WREG	WREG = 对 f 执行带进位的循环左移	1	1	C、 N 和 Z
	RLC	Ws,Wd	Wd = 对 Ws 执行带进位的循环左移	1	1	C、 N 和 Z
RLNC	RLNC	f	f = 循环左移 f （不带进位）	1	1	N 和 Z
	RLNC	f,WREG	WREG = 循环左移 f （不带进位）	1	1	N 和 Z
	RLNC	Ws,Wd	Wd = 循环左移 Ws （不带进位）	1	1	N 和 Z
RRC	RRC	f	f = 对 f 执行带进位的循环右移	1	1	C、 N 和 Z
	RRC	f,WREG	WREG = 对 f 执行带进位的循环右移	1	1	C、 N 和 Z
	RRC	Ws,Wd	Wd = 对 Ws 执行带进位的循环右移	1	1	C、 N 和 Z
RRNC	RRNC	f	f = 循环右移 f （不带进位）	1	1	N 和 Z
	RRNC	f,WREG	WREG = 循环右移 f （不带进位）	1	1	N 和 Z
	RRNC	Ws,Wd	Wd = 循环右移 Ws （不带进位）	1	1	N 和 Z
SE	SE	Ws,Wnd	Wnd = 符号扩展后的 Ws	1	1	C、 N 和 Z
SETM	SETM	f	f = FFFFh	1	1	无
	SETM	WREG	WREG = FFFFh	1	1	无
	SETM	Ws	Ws = FFFFh	1	1	无
SL	SL	f	f = 左移 f	1	1	C、 N、 OV 和 Z
	SL	f,WREG	WREG = 左移 f	1	1	C、 N、 OV 和 Z
	SL	Ws,Wd	Wd = 左移 Ws	1	1	C、 N、 OV 和 Z
	SL	Wb,Wns,Wnd	Wnd = 将 Wb 左移 Wns 位	1	1	N 和 Z
	SL	Wb,#lit5,Wnd	Wnd = 将 Wb 左移 lit5 位	1	1	N 和 Z
SUB	SUB	f	f = f - WREG	1	1	C、DC、N、OV和Z
	SUB	f,WREG	WREG = f - WREG	1	1	C、DC、N、OV和Z
	SUB	#lit10,Wn	Wn = Wn - lit10	1	1	C、DC、N、OV和Z
	SUB	Wb,Ws,Wd	Wd = Wb - Ws	1	1	C、DC、N、OV和Z
	SUB	Wb,#lit5,Wd	Wd = Wb - lit5	1	1	C、DC、N、OV和Z
SUBB	SUBB	f	f = f - WREG -($\overline{C}$)	1	1	C、DC、N、OV和Z
	SUBB	f,WREG	WREG = f - WREG -($\overline{C}$)	1	1	C、DC、N、OV和Z
	SUBB	#lit10,Wn	Wn = Wn - lit10 -($\overline{C}$)	1	1	C、DC、N、OV和Z
	SUBB	Wb,Ws,Wd	Wd = Wb - Ws - ($\overline{C}$)	1	1	C、DC、N、OV和Z
	SUBB	Wb,#lit5,Wd	Wd = Wb - lit5 - ($\overline{C}$)	1	1	C、DC、N、OV和Z
SUBR	SUBR	f	f = WREG - f	1	1	C、DC、N、OV和Z
	SUBR	f,WREG	WREG = WREG - f	1	1	C、DC、N、OV和Z
	SUBR	Wb,Ws,Wd	Wd = Ws - Wb	1	1	C、DC、N、OV和Z
	SUBR	Wb,#lit5,Wd	Wd = lit5 - Wb	1	1	C、DC、N、OV和Z
SUBBR	SUBBR	f	f = WREG - f -($\overline{C}$)	1	1	C、DC、N、OV和Z
	SUBBR	f,WREG	WREG = WREG - f - ($\overline{C}$)	1	1	C、DC、N、OV和Z
	SUBBR	Wb,Ws,Wd	Wd = Ws - Wb - ($\overline{C}$)	1	1	C、DC、N、OV和Z
	SUBBR	Wb,#lit5,Wd	Wd = lit5 - Wb - ($\overline{C}$)	1	1	C、DC、N、OV和Z
SWAP	SWAP.b	Wn	Wn = 半字节交换 Wn 内容	1	1	无
	SWAP	Wn	Wn = 将 Wn 的两个字节相交换	1	1	无

表 26-2： 指令集汇总 （续）

汇编指令助记符	汇编语法		说明	字数	周期数	受影响的状态标志位
TBLRDH	TBLRDH	Ws,Wd	将程序存储单元的 <23:16> 读入 Wd<7:0>	1	2	无
TBLRDL	TBLRDL	Ws,Wd	将程序存储单元的 <15:0> 读入 Wd	1	2	无
TBLWTH	TBLWTH	Ws,Wd	将 Ws<7:0> 写入程序存储单元的 <23:16>	1	2	无
TBLWTL	TBLWTL	Ws,Wd	将 Ws 写入程序存储单元的 <15:0>	1	2	无
ULNK	ULNK		释放堆栈帧	1	1	无
XOR	XOR	f	f = f .XOR. WREG	1	1	N 和 Z
	XOR	f,WREG	WREG = f .XOR. WREG	1	1	N 和 Z
	XOR	#lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N 和 Z
	XOR	Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N 和 Z
	XOR	Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N 和 Z
ZE	ZE	Ws,Wnd	Wnd = 零扩展后的 Ws	1	1	C、 N 和 Z

注：

27.0 电气特性

本节概述了 PIC24FJ256GA110 系列的电气规范。其余信息将在该文档的后续版本中给出。

PIC24FJ256GA110 系列的绝对最大值如下所列。器件长时间工作在最大额定值条件可能影响其可靠性。我们建议不要使器件在该规范规定的参数范围外工作。

绝对最大值 (†)

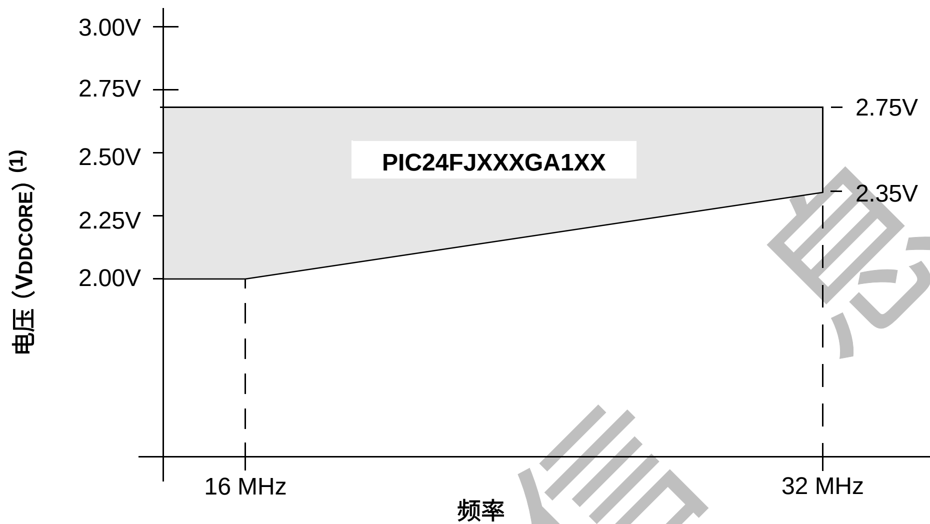
环境温度	-40°C 至 +100°C
存储温度	-65°C 至 +150°C
V_{DD} 引脚相对于 V_{SS} 的电压	-0.3V 至 +4.0V
任一模拟和数字引脚以及 $\overline{MCLR}$ 引脚相对于 V_{SS} 的电压	-0.3V 至 （V_{DD} + 0.3V）
仅数字引脚相对于 V_{SS} 的电压	-0.3V 至 +6.0V
V_{DDCORE} 引脚相对于 V_{SS} 的电压	-0.3V 至 +3.0V
流出 V_{SS} 引脚的最大电流	300 mA
流入 V_{DD} 引脚的最大电流 （**注 1**）	250 mA
任一 I/O 引脚的最大输出灌电流	25 mA
任一 I/O 引脚的最大输出拉电流	25 mA
所有端口的最大灌电流	200 mA
所有端口的最大拉电流 （**注 1**）	200 mA

注 1：允许的最大电流由器件的最大功耗决定 （见表 27-1）。

† 注：如果器件工作条件超过上述 “ 绝对最大值 ”，可能引起器件永久性损坏。这仅是极限参数，我们不建议器件工作在极限值甚至超过上述极限值。器件长时间工作在极限条件下可能会影响其可靠性。

27.1 直流特性

图 27-1： PIC24FJ256GA110 系列电压—频率关系图 （工业级）



对于 16 MHz 至 32 MHz 的频率， FMAX = (45.714 MHz/V) * (VDDCORE 2V) + 16 MHz。

注 1： 如果禁用稳压器，则 VDD 和 VDDCORE 必须始终满足以下条件：VDDCORE ≤ VDD ≤ 3.6V。

表 27-1： 热工作条件

参数	符号	最小值	典型值	最大值	单位
PIC24FJ256GA110 系列：					
工作结温范围	T_J	-40	—	+125	°C
工作环境温度范围	T_A	-40	—	+85	°C
功耗： 芯片内部功耗 $P_{INT} = V_{DD} \times (I_{DD} - \Sigma I_{OH})$ I/O 引脚功耗： $P_{I/O} = \Sigma (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma (V_{OL} \times I_{OL})$	P_D	$P_{INT} + P_{I/O}$			W
允许的最大功耗	P_{DMAX}	$(T_J - T_A)/\theta_{JA}$			W

表 27-2： 热封装特性

特性	符号	典型值	最大值	单位	注
封装热阻， 14x14x1 mm TQFP	θ_{JA}	50.0	—	°C/W	(注 1)
封装热阻， 12x12x1 mm TQFP	θ_{JA}	69.4	—	°C/W	(注 1)
封装热阻， 10x10x1 mm TQFP	θ_{JA}	76.6	—	°C/W	(注 1)

注 1： 通过封装模拟获得结点与环境的热阻值 （θ_{JA}）。

表 27-3： DC 特性：温度和电压规范

DC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C				
参数编号	**符号**	**特性**	**最小值**	**典型值 (1)**	**最大值**	**单位**	**条件**
工作电压							
DC10	**电源电压**						
	VDD		2.2	—	3.6	V	使能稳压器
	VDD		VDDCORE	—	3.6	V	禁止稳压器
	VDDCORE		2.0	—	2.75	V	禁止稳压器
DC12	VDR	**RAM 数据保持电压 (2)**	1.5	—	—	V	
DC16	VPOR	确保内部上电复位信号的 **VDD 起始电压**	—	VSS	—	V	
DC17	SVDD	确保内部上电复位信号的 **VDD 上升率**	0.05	—	—	V/ms	0.1s 内电压变化范围为 0-3.3V 60 ms 内电压变化范围为 0-2.5V

注 1： 除非另外声明，否则 “ 典型值 ” 栏中的数据都是在 3.3V、 25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

2： 这是在不丢失 RAM 数据的前提下的 VDD 的下限值。

表 27-4： DC 特性：工作电流 （IDD）

DC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C			
参数编号	典型值 (1)	最大值	单位	条件		
工作电流 （IDD） (2)						
DC20	1.0	TBD	mA	-40°C	2.0V(3)	1 MIPS
DC20a	1.0	TBD	mA	+25°C		
DC20b	1.0	TBD	mA	+85°C		
DC20d	1.8	TBD	mA	-40°C	3.3V(4)	
DC20e	1.8	TBD	mA	+25°C		
DC20f	1.8	TBD	mA	+85°C		
DC23	4.0	TBD	mA	-40°C	2.0V(3)	4 MIPS
DC23a	4.0	TBD	mA	+25°C		
DC23b	4.0	TBD	mA	+85°C		
DC23d	6.3	TBD	mA	-40°C	3.3V(4)	
DC23e	6.3	TBD	mA	+25°C		
DC23f	6.3	TBD	mA	+85°C		
DC24	20.7	TBD	mA	-40°C	2.5V(3)	16 MIPS
DC24a	20.7	TBD	mA	+25°C		
DC24b	20.7	TBD	mA	+85°C		
DC24d	23.0	TBD	mA	-40°C	3.3V(4)	
DC24e	23.0	TBD	mA	+25°C		
DC24f	23.0	TBD	mA	+85°C		
DC31	13.0	TBD	μA	-40°C	2.0V(3)	LPRC （31 kHz）
DC31a	13.0	TBD	μA	+25°C		
DC31b	20.0	TBD	μA	+85°C		
DC31d	54.0	TBD	μA	-40°C	3.3V(4)	
DC31e	54.0	TBD	μA	+25°C		
DC31f	95.0	TBD	μA	+85°C		

图注： TBD = 待定

注 1： 除非另外声明，否则 “ 典型值 ” 栏中的数据都是在 3.3V、 25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

2： 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关频率、振荡器类型、内部代码执行模式以及温度也对电流消耗有影响。所有 IDD 测量的测试条件为：使用满幅的外部方波驱动 OSC1。所有 I/O 引脚配置为输入且被拉到 VDD。
MCLR = VDD， WDT 和 FSCM 被禁止。 CPU、 SRAM、程序存储器和数据存储器处于工作状态。外设模块均不工作且所有的外设模块禁止 （PMD）位均置 1。

3： 禁止片内稳压器 （ENVREG 连到 VSS）。

4： 使能片上稳压器 （ENVREG 连接到 VDD），使能低压检测 （LVD）和欠压检测 （BOD）。

表 27-5： DC 特性：空闲电流 （I_{IDLE}）

DC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ T_A ≤ +85°C			
参数编号	典型值 (1)	最大值	单位	条件		
空闲电流 （I_{IDLE}） (2)						
DC40	150	TBD	µA	-40°C	2.0V(3)	1 MIPS
DC40a	150	TBD	µA	+25°C		
DC40b	150	TBD	µA	+85°C		
DC40d	250	TBD	µA	-40°C	3.3V(4)	
DC40e	250	TBD	µA	+25°C		
DC40f	250	TBD	µA	+85°C		
DC43	0.55	TBD	mA	-40°C	2.0V(3)	4 MIPS
DC43a	0.55	TBD	mA	+25°C		
DC43b	0.55	TBD	mA	+85°C		
DC43d	0.82	TBD	mA	-40°C	3.3V(4)	
DC43e	0.82	TBD	mA	+25°C		
DC43f	0.82	TBD	mA	+85°C		
DC47	3.0	TBD	mA	-40°C	2.5V(3)	16 MIPS
DC47a	3.0	TBD	mA	+25°C		
DC47b	3.0	TBD	mA	+85°C		
DC47c	3.5	TBD	mA	-40°C	3.3V(4)	
DC47d	3.5	TBD	mA	+25°C		
DC47e	3.5	TBD	mA	+85°C		
DC50	0.85	TBD	mA	-40°C	2.0V(3)	FRC （4 MIPS）
DC50a	0.85	TBD	mA	+25°C		
DC50b	0.85	TBD	mA	+85°C		
DC50d	1.2	TBD	mA	-40°C	3.3V(4)	
DC50e	1.2	TBD	mA	+25°C		
DC50f	1.2	TBD	mA	+85°C		
DC51	4.0	TBD	µA	-40°C	2.0V(3)	LPRC （31 kHz）
DC51a	4.0	TBD	µA	+25°C		
DC51b	7.0	TBD	µA	+85°C		
DC51d	42	TBD	µA	-40°C	3.3V(4)	
DC51e	42	TBD	µA	+25°C		
DC51f	70	TBD	µA	+85°C		

图注： TBD = 待定

注 1： 除非另外声明，否则 “ 典型值 ” 栏中的数据都是在 3.3V、 25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

2： 基本 I_{IDLE} 电流测量是在内核不工作、使用满幅的外部方波驱动 OSC1 的情况下发生。所有 I/O 引脚设置为输入并拉至 V_{DD}。 $\overline{MCLR}$ = V_{DD}， WDT 和 FSCM 被禁止。外设模块均不工作且所有的外设模块禁止 （PMD）位均置 1。

3： 禁止片内稳压器 （ENVREG 连到 V_{SS}）。

4： 使能片上稳压器 （ENVREG 连接到 V_{DD}），使能低压检测 （LVD）和欠压检测 （BOD）。

表 27-6： DC 特性：掉电电流 （IPD）

DC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C			
参数编号	**典型值 (1)**	**最大值**	**单位**	**条件**		
掉电电流 （IPD） (2)						
DC60	0.1	TBD	μA	-40°C	2.0V(3)	基本掉电电流 (5)
DC60a	0.15	TBD	μA	+25°C		
DC60b	3.7	TBD	μA	+85°C		
DC60c	0.2	TBD	μA	-40°C	2.5V(3)	
DC60d	0.25	TBD	μA	+25°C		
DC60e	4.2	TBD	μA	+85°C		
DC60f	3.3	TBD	μA	-40°C	3.3V(4)	
DC60g	3.5	TBD	μA	+25°C		
DC60h	9.0	TBD	μA	+85°C		
DC61	1.75	3	μA	-40°C	2.0V(3)	看门狗定时器电流：ΔIWDT(5)
DC61a	1.75	3	μA	+25°C		
DC61b	1.75	3	μA	+85°C		
DC61c	2.4	4	μA	-40°C	2.5V(3)	
DC61d	2.4	4	μA	+25°C		
DC61e	2.4	4	μA	+85°C		
DC61f	2.8	5	μA	-40°C	3.3V(4)	
DC61g	2.8	5	μA	+25°C		
DC61h	2.8	5	μA	+85°C		
DC62	5.0	TBD	μA	-40°C	2.0V(3)	RTCC 和使用 32 kHz 晶振的 Timer1：ΔRTCC + ΔITI32(5)
DC62a	8.0	TBD	μA	+25°C		
DC62b	8.0	TBD	μA	+85°C		
DC62c	6.0	TBD	μA	-40°C	2.5V(3)	
DC62d	9.0	TBD	μA	+25°C		
DC62e	9.0	TBD	μA	+85°C		
DC62f	6.5	TBD	μA	-40°C	3.3V(4)	
DC62g	9.5	TBD	μA	+25°C		
DC62h	10.0	TBD	μA	+85°C		

图注： TBD = 待定

注 1： 除非另外声明，否则 “ 典型值 ” 栏中的数据都是在 3.3V、 25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

2： 基本 IPD 是在关闭所有外设和时钟的条件下测得的。所有 I/O 引脚都被配置为输入且被拉至高电平。 WDT 等外设均被关闭，且所有的外设模块禁止 （PMD）位均置 1。

3： 禁止片内稳压器 （ENVREG 连接到 VSS）。

4： 使能片内稳压器 （ENVREG 连接到 VDD），使能低压检测 （LVD）和欠压检测 （BOD）。

5： Δ 电流为模块使能时额外消耗的电流。掉电时外设模块的电流消耗是这一电流与基本 IPD 电流之和。

表 27-7： DC 特性：I/O 引脚输入规范

DC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C				
参数编号	符号	特性	最小值	典型值 [1]	最大值	单位	条件
	VIL	**输入低电平电压 [4]**					
DI10		具有 ST 缓冲器的 I/O 引脚	VSS	—	0.2 VDD	V	
DI11		具有 TTL 缓冲器的 I/O 引脚	VSS	—	0.15 VDD	V	
DI15		$\overline{\text{MCLR}}$	VSS	—	0.2 VDD	V	
DI16		OSC1 （XT 模式）	VSS	—	0.2 VDD	V	
DI17		OSC1 （HS 模式）	VSS	—	0.2 VDD	V	
DI18		具有 I²C™ 缓冲器的 I/O 引脚	VSS	—	0.3 VDD	V	
DI19		具有 SMBus 缓冲器的 I/O 引脚	VSS	—	0.8	V	使能 SMBus
	VIH	**输入低电平电压 [4]**					
DI20		具有 ST 缓冲器的 I/O 引脚：					
		具有模拟功能	0.8 VDD	—	VDD	V	
		仅有数字功能	0.8 VDD	—	5.5	V	
DI21		具有 TTL 缓冲器的 I/O 引脚：					
		具有模拟功能	0.25 VDD + 0.8	—	VDD	V	
		仅有数字功能	0.25 VDD + 0.8	—	5.5	V	
DI25		$\overline{\text{MCLR}}$	0.8 VDD	—	VDD	V	
DI26		OSC1 （XT 模式）	0.7 VDD	—	VDD	V	
DI27		OSC1 （HS 模式）	0.7 VDD	—	VDD	V	
DI28		具有 I²C 缓冲器的 I/O 引脚：					
		具有模拟功能	0.7 VDD	—	VDD	V	
		仅有数字功能	0.7 VDD	—	5.5	V	
DI29		具有 SMBus 缓冲器的 I/O 引脚：					2.5V ≤ VPIN ≤ VDD
		具有模拟功能	2.1		VDD	V	
		仅有数字功能	2.1		5.5	V	
DI30	ICNPU	**CNxx 上拉电流**	50	250	400	μA	VDD = 3.3V，VPIN = VSS
	IIL	**输入泄漏电流 [2,3]**					
DI50		I/O 端口	—	—	±1	μA	VSS ≤ VPIN ≤ VDD，引脚处于高阻态
DI51		模拟输入引脚	—	—	±1	μA	VSS ≤ VPIN ≤ VDD，引脚处于高阻态
DI55		$\overline{\text{MCLR}}$	—	—	±1	μA	VSS ≤ VPIN ≤ VDD
DI56		OSC1	—	—	±1	μA	VSS ≤ VPIN ≤ VDD，XT 和 HS 模式

注 1： 除非另外声明，否则 “ 典型值 ” 栏中的数据都是在 3.3V、 25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

2： $\overline{\text{MCLR}}$ 引脚上的泄漏电流主要取决于所加电平。表中给定的电平表示正常工作条件下的电平。在不同输入电压条件下可能测得更高的泄漏电流。

3： 负电流定义为引脚的拉电流。

4： 关于 I/O 引脚缓冲器的类型请参见表 1-4。

表 27-8： DC 特性：I/O 引脚输出规范

DC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C				
参数编号	符号	特性	最小值	典型值[1]	最大值	单位	条件
	VOL	**输出低电平电压**					
DO10		I/O 端口	—	—	0.4	V	IOL = 8.5 mA， VDD = 3.6V
			—	—	0.4	V	IOL = 6.0 mA, VDD = 2.0V
DO16		OSC2/CLKO	—	—	0.4	V	IOL = 8.5 mA， VDD = 3.6V
			—	—	0.4	V	IOL = 6.0 mA, VDD = 2.0V
	VOH	**输出高电平电压**					
DO20		I/O 端口	3.0	—	—	V	IOH = -3.0 mA， VDD = 3.6V
			2.4	—	—	V	IOH = -6.0 mA， VDD = 3.6V
			1.65	—	—	V	IOH = -1.0 mA， VDD = 2.0V
			1.4	—	—	V	IOH = -3.0 mA， VDD = 2.0V
DO26		OSC2/CLKO	2.4	—	—	V	IOH = -6.0 mA， VDD = 3.6V
			1.4	—	—	V	IOH = -3.0 mA， VDD = 2.0V

注 1： 除非另外声明，否则 “ 典型值 ” 栏中的数据都是在 3.3V、 25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

表 27-9： DC 特性：程序存储器

DC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C				
参数编号	符号	特性	最小值	典型值[1]	最大值	单位	条件
		闪存程序存储器					
D130	EP	单元耐擦写能力	10000	—	—	E/W	-40°C 至 +85°C
D131	VPR	读操作时的 VDD	VMIN	—	3.6	V	VMIN = 最小工作电压
D132B	VPEW	自定时写操作时的 VDD	2.25	—	3.6	V	VMIN = 最小工作电压
D133A	TIW	自定时写周期时间	—	3	—	ms	
D133B	TIE	自定时页擦除时间	40	—	—	ms	
D134	TRETD	特性保持时间	20	—	—	年	假定未违反其他规范
D135	IDDP	编程时的供电电流	—	7	—	mA	

注 1： 除非另外声明，否则 “ 典型值 ” 栏中的数据都是在 3.3V、 25°C 的条件下给出的。

表 27-10： 内部稳压器规范

工作条件：-40°C < TA < +85°C （除非另外声明）							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
	VRGOUT	稳压器输出电压	—	2.5	—	V	
	CEFC	外部滤波电容值	4.7	10	—	μF	建议该电容的等效串联阻抗 < 3 欧姆；必须 < 5 欧姆。
	TVREG		—	50	—	μs	ENVREG 连接到 VDD
	TPWRT		—	64	—	ms	ENVREG 连接到 VSS

27.2 AC 特性和时序参数

本节包含的信息说明了 PIC24FJ256GA110 系列器件的 AC 特性和时序参数。

表 27-11： 温度和电压规范—AC

AC 特性	**标准工作条件：2.0V 至 3.6V （除非另外声明）** 工作温度 （工业级：-40°C ≤ T_A ≤ +85°C） 工作电压 V_{DD} 范围如**第 27.1 节 " 直流特性 "** 所示。

图 27-2： 器件时序规范的负载条件

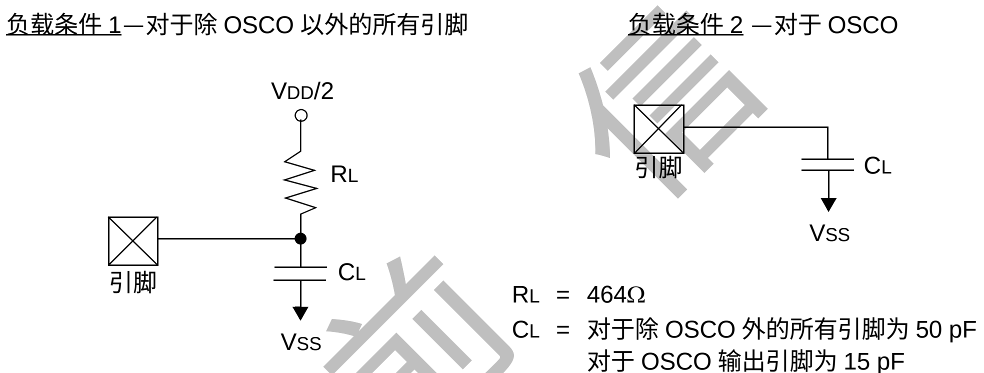


表 27-12： 输出引脚上的容性负载要求

参数编号	符号	特性	最小值	典型值 (1)	最大值	单位	条件
DO50	C_{OSC2}	OSCO/CLKO 引脚	—	—	15	pF	当外部时钟用于驱动 OSCI 时，处于 XT 和 HS 模式。
DO56	C_{IO}	所有 I/O 引脚和 OSCO	—	—	50	pF	EC 模式。
DO58	C_B	SCLx 和 SDAx	—	—	400	pF	在 I²C™ 模式下。

注 1： 除非另外声明，否则 " 典型值 " 栏中的数据都是在 3.3V、 25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

图 27-3： 外部时钟时序

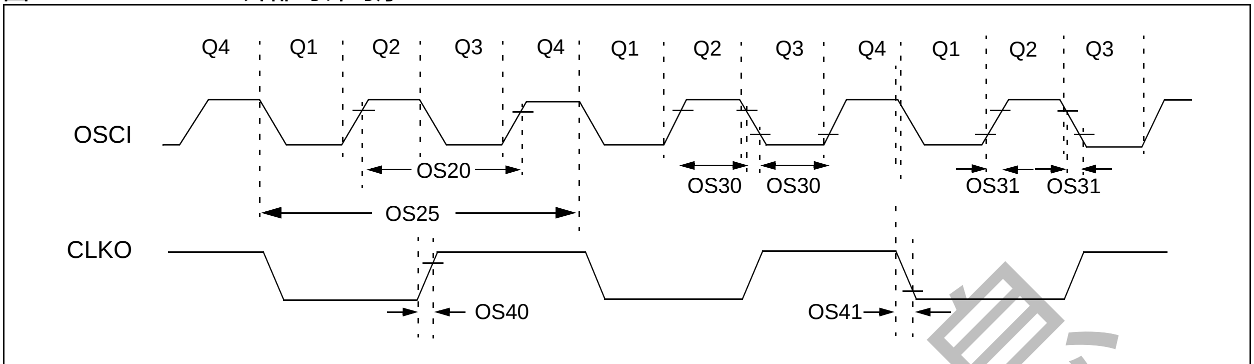


表 27-13： 外部时钟时序要求

AC 特性			标准工作条件：2.50 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C				
参数编号	符号	特性	最小值	典型值 (1)	最大值	单位	条件
OS10	FOSC	外部 CLKI 频率 （外部时钟仅允许运行于 EC 模式）	DC 4	— —	32 8	MHz MHz	EC ECPLL
		振荡器频率	3 4 10 31	— — — —	10 8 32 33	MHz MHz MHz kHz	XT XTPLL HS SOSC
OS20	TOSC	TOSC = 1/FOSC	—	—	—	—	关于 FOSC 值请参见参数 OS10
OS25	TCY	指令周期 (2)	62.5	—	DC	ns	
OS30	TosL, TosH	外部时钟输入 （OSCI） 高电平或低电平时间	0.45 x TOSC	—	—	ns	EC
OS31	TosR, TosF	外部时钟输入 （OSCI） 上升或下降时间	—	—	20	ns	EC
OS40	TckR	CLKO 上升时间 (3)	—	6	10	ns	
OS41	TckF	CLKO 下降时间 (3)	—	6	10	ns	

注 1： 除非另外声明，否则 “ 典型值 ” 栏中的数据都是在 3.3V、 25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

2： 指令周期 （TCY）等于输入振荡器时基周期的两倍。所有规定值均基于标准运行条件下，器件执行代码时对应特定振荡器类型的特征数据。超过规定值可导致振荡器运行不稳定，和 / 或使电流消耗超过预期。所有器件在测试 “ 最小 ” 值时，均在 OSCI/CLKI 引脚接入了外部时钟。当使用外部时钟输入时，所有器件的 “ 最大 ” 周期时间限制为 “ DC ”（无时钟）。

3： 测量在 EC 模式下进行。在 OSCO 引脚上测量 CLKO 信号。 CLKO 在 Q1-Q2 周期 （1/2 TCY）中为低电平，在 Q3-Q4 周期 （1/2 TCY）中为高电平。

表 27-14： PLL 时钟时序规范 （VDD = 2.0V 至 3.6V）

AC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C				
参数编号	符号	特性 [1]	最小值	典型值 [2]	最大值	单位	条件
OS50	FPLLI	PLL 输入频率范围 [2]	4	—	8	MHz	ECPLL、 HSPLL 和 XTPLL 模式
OS51	FSYS	片上 VCO 系统频率	16	—	32	MHz	
OS52	TLOCK	PLL 启动时间 （锁定时间）	—	—	2	ms	
OS53	DCLK	CLKO 稳定性 （抖动性）	-2	1	+2	%	

注 1： 这些参数为特性值，但生产时未经测试。
2： 除非另外声明，否则 “ 典型值 ” 栏中的数据都是在 3.3V、 25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

表 27-15： AC 特性：内部 RC 精度

AC 特性		标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C					
参数编号	特性	最小值	典型值	最大值	单位	条件	
	内部 FRC 精度 @ 8 MHz[1]						
F20	**FRC**	-2	—	2	%	+25°C	3.0V ≤ VDD ≤ 3.6V
		-5	—	5	%	-40°C ≤ TA ≤ +85°C	3.0V ≤ VDD ≤ 3.6V

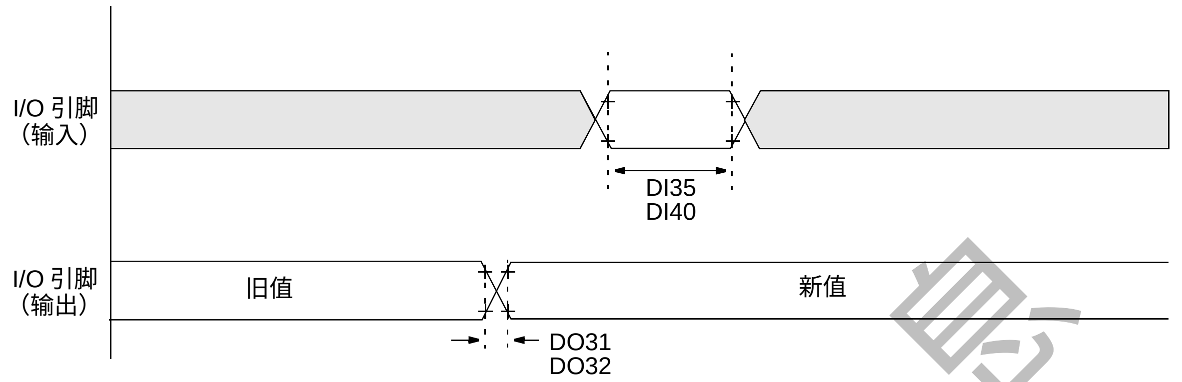
注 1： 已在 25°C、 3.3V 条件下进行了频率校准。 OSCTUN 位可用来补偿温度漂移。

表 27-16： 内部 RC 精度

AC 特性		标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C					
参数编号	特性	最小值	典型值	最大值	单位	条件	
	LPRC @ 31 kHz[1]						
F21		-20	—	20	%	-40°C ≤ TA ≤ +85°C	3.0V ≤ VDD ≤ 3.6V

注 1： LPRC 频率将随 VDD 的变化而变化。

图 27-4：　　CLKO 和 I/O 时序特性



注：负载条件请参见图 27-2。

表 27-17：　　CLKO 和 I/O 时序要求

AC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C				
参数编号	符号	特性	最小值	典型值 (1)	最大值	单位	条件
DO31	TIOR	端口输出上升时间	—	10	25	ns	
DO32	TIOF	端口输出下降时间	—	10	25	ns	
DI35	TINP	INTx 引脚高电平或低电平时间 （输出）	20	—	—	ns	
DI40	TRBP	CNx 高电平或低电平时间 （输入）	2	—	—	TCY	

注　1： 除非另外声明，否则 “ 典型值 ” 栏中的数据都是在 3.3V、 25°C 的条件下给出的。

表 27-18： ADC 模块规范

AC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C				
参数编号	**符号**	**特性**	**最小值**	**典型值**	**最大值**	**单位**	**条件**
器件电源							
AD01	AVDD	模块电源 VDD	取 VDD – 0.3 或 2.0 中的较大值	—	取 VDD + 0.3 或 3.6 中的较小值	V	
AD02	AVSS	模块电源 VSS	VSS – 0.3	—	VSS + 0.3	V	
参考输入							
AD05	VREFH	参考电压高电平	AVSS + 1.7	—	AVDD	V	
AD06	VREFL	参考电压低电平	AVSS	—	AVDD – 1.7	V	
AD07	VREF	绝对参考电压	AVSS – 0.3	—	AVDD + 0.3	V	
模拟输入							
AD10	VINH-VINL	满量程输入范围	VREFL	—	VREFH	V	（注 2）
AD11	VIN	绝对输入电压	AVSS – 0.3	—	AVDD + 0.3	V	—
AD12	VINL	绝对 VINL 输入电压	AVSS – 0.3		AVDD/2	V	
AD17	RIN	模拟信号源的推荐阻抗	—	—	2.5K	Ω	10 位
ADC 精度							
AD20b	Nr	分辨率	—	10	—	位	
AD21b	INL	积分非线性误差	—	±1	<±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22b	DNL	微分非线性误差	—	±0.5	<±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23b	GERR	增益误差	—	±1	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24b	EOFF	失调误差	—	±1	±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25b	—	单调性 (1)	—	—	—	—	保证

注 1： ADC 转换结果将不会随着输入电压的增加而减小，而且不会丢失代码。
2： 测量是在使用外部 VREF+ 和 VREF- 作为 ADC 参考电压的情况下进行的。

表 27-19： ADC 转换时序要求 (1)

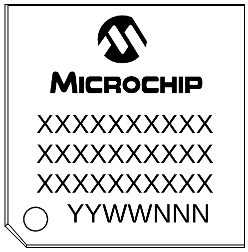
AC 特性			标准工作条件：2.0V 至 3.6V （除非另外声明） 工作温度 （工业级）： -40°C ≤ TA ≤ +85°C				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
时钟参数							
AD50	TAD	ADC 时钟周期	75	—	—	ns	TCY = 75 ns， AD1CON3 处于默认状态
AD51	tRC	ADC 内部 RC 振荡器周期	—	250	—	ns	
转换速率							
AD55	tCONV	转换时间	—	12	—	TAD	
AD56	FCNV	吞吐率	—	—	500	ksps	AVDD > 2.7V
AD57	tSAMP	采样时间	—	1	—	TAD	
时钟参数							
AD61	tPSS	将采样位 （SAMP）置 1 到采样启动之间的延时	2	—	3	TAD	

注 1： 因为采样电容最终将无法保持电荷，因此低于 10 kHz 的时钟频率可能影响线性性能，尤其是在温度较高时。

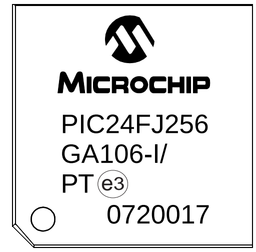
28.0 封装信息

28.1 封装标识信息

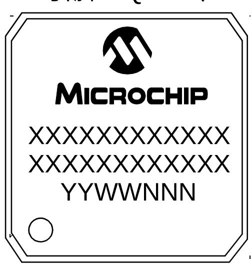
64 引脚 TQFP（10x10x1 mm）



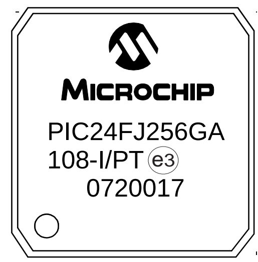
示例



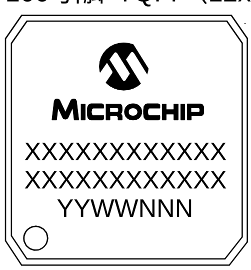
80 引脚 TQFP（12x12x1 mm）



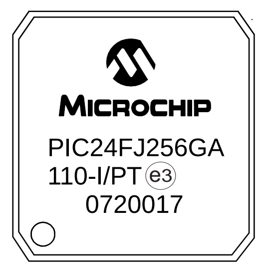
示例



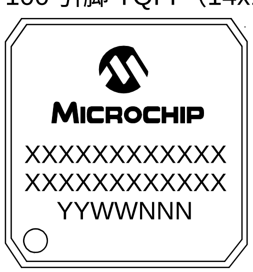
100 引脚 TQFP（12x12x1 mm）



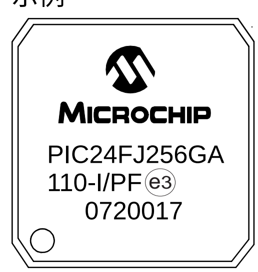
示例



100 引脚 TQFP（14x14x1 mm）



示例



图注：

XX...X	客户信息
Y	年份代码 （公历年的最后一位数字）
YY	年份代码 （公历年的最后两位数字）
WW	星期代码 （一月一日的星期代码为 “01”）
NNN	以字母数字排序的追踪代码
e3	雾锡 （Matte Tin， Sn）的 JEDEC 无铅标志
*	表示无铅封装。 JEDEC 无铅标志 （e3）标示于此种封装的外包装上。

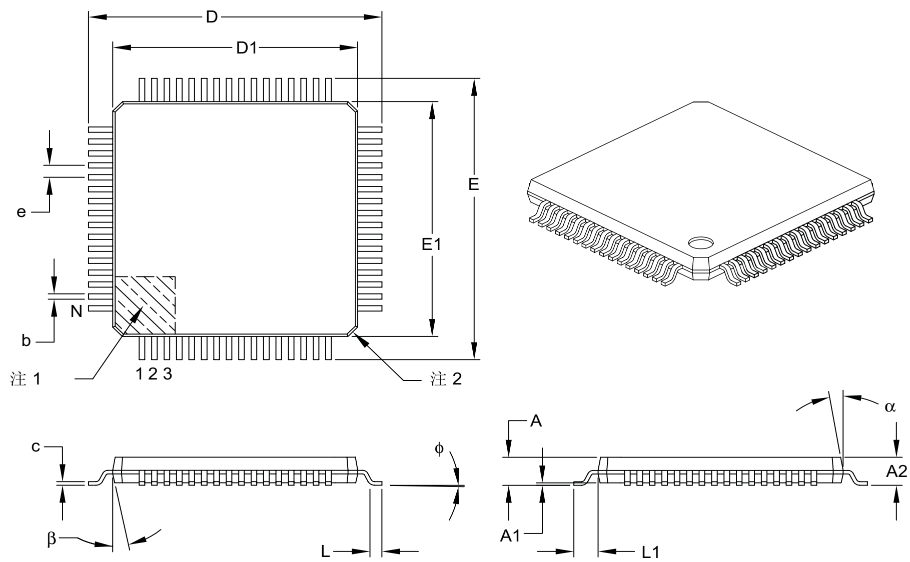
注： Microchip 元器件编号如果无法在同一行内完整标注，将换行标出，因此会限制表示客户信息的字符数。

28.2 封装详细信息

以下部分将介绍各种封装的技术细节。

64 引脚塑封薄型四方扁平封装（PT）—— 主体 10 x 10 x 1 mm，2.00 mm [TQFP]

注：最新封装图请至 http://www.microchip.com/packaging 查看 Mcirochip 封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N		64	
引脚间距	e		0.50 BSC	
总高度	A	–	–	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	–	0.15
底脚长度	L	0.45	0.60	0.75
底脚占位长度	L1		1.00 REF	
底脚倾斜角	φ	0°	3.5°	7°
总宽度	E		12.00 BSC	
总长度	D		12.00 BSC	
塑模封装宽度	E1		10.00 BSC	
塑模封装长度	D1		10.00 BSC	
引脚厚度	c	0.09	–	0.20
引脚宽度	b	0.17	0.22	0.27
塑模顶部锥度	α	11°	12°	13°
塑模底部锥度	β	11°	12°	13°

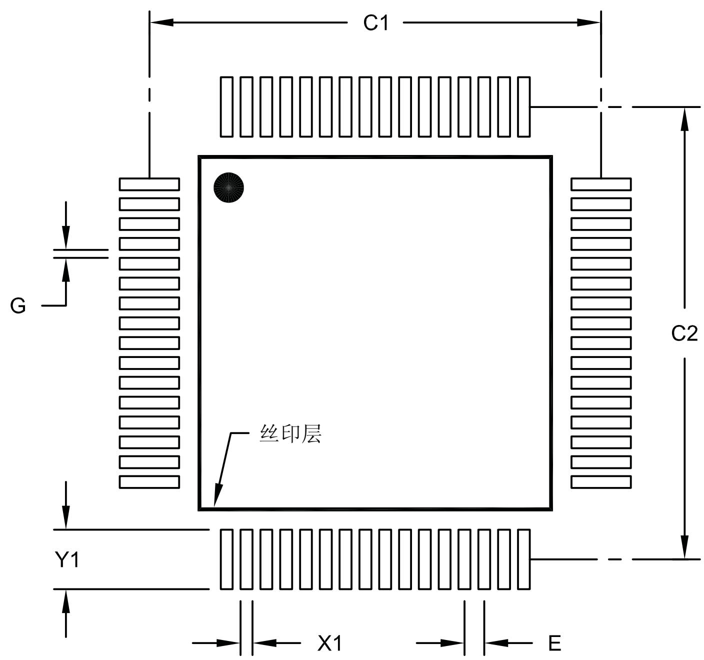
注：
1. 引脚1定位特性可能有变化，但一定位于阴影区域内。
2. 拐角处可能存在斜面，尺寸可变。
3. 尺寸D1和E1不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.25 mm。
4. 尺寸和公差遵循ASME Y14.5M。
   BSC：基本尺寸。理论精确值，不含公差。
   REF：参考尺寸。通常不包含公差，仅供参考。

Microchip Technology 图号C04-085B

 超前信息

64 引脚塑封薄型四方扁平封装 （PT）—— 主体 10 x 10 x 1 mm， 2.00 mm [TQFP]

注：最新封装图请至 http://www.microchip.com/packaging 查看 Mcirochip 封装规范。



建议的焊盘布局

单位		毫米		
尺寸范围		最小	正常	最大
触点间距	E		0.5 BSC	
触点焊盘间距	C1		11.40	
触点焊盘间距	C2		11.40	
触点焊盘宽度（X64）	X1			0.30
触点焊盘长度（X64）	Y1			1.50
焊盘间距	G	0.20		

注：

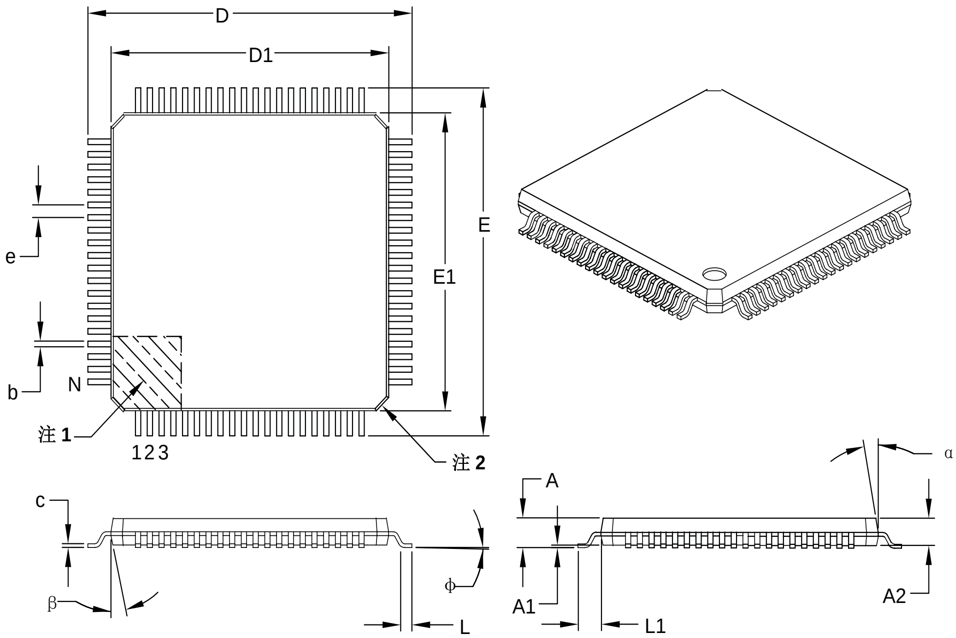
1. 尺寸和公差遵循ASME Y14.5M

BSC：基本尺寸。理论精确值，不含公差。

Microchip Technology 图号 C04-2085A

80 引脚塑封薄型四方扁平封装（PT）—— 主体 12 x 12 x 1 mm，2.00 mm [TQFP]

注：最新封装图请至 http://www.microchip.com/packaging 查看 Mcirochip 封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N	80		
引脚间距	e	0.50 BSC		
总高度	A	–	–	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	–	0.15
底脚长度	L	0.45	0.60	0.75
底脚占位长度	L1	1.00 REF		
底脚倾斜角度	φ	0°	3.5°	7°
总宽度	E	14.00 BSC		
总长度	D	14.00 BSC		
塑模封装厚度	E1	12.00 BSC		
塑模封装长度	D1	12.00 BSC		
引脚厚度	c	0.09	–	0.20
引脚宽度	b	0.17	0.22	0.27
塑模顶部锥度	α	11°	12°	13°
塑模底部锥度	β	11°	12°	13°

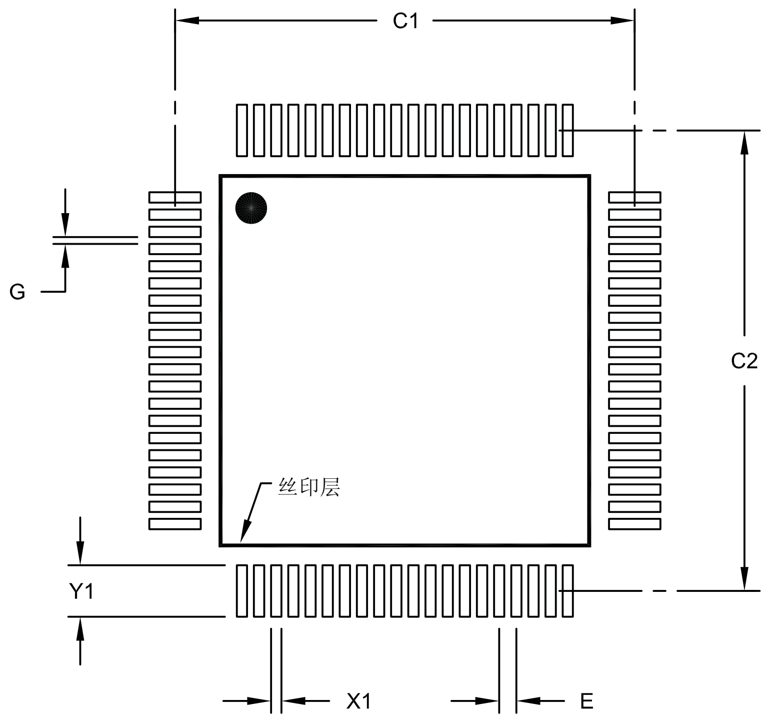
注：

1. 引脚 1 可视定位标记可能会有变化，但一定位于阴影区域内。
2. 拐角处可能存在斜面；尺寸可变。
3. 尺寸 D1 和 E1 不包含塑模的毛边或突起。塑模每侧的毛边或突起不应超过 0.25 mm。
4. 尺寸和公差遵循 ASME Y 14.5M。
   BSC：基本尺寸。理论精确值，不含公差。
   REF：参考尺寸。通常不包含公差，仅供参考。

Microchip Technology 图号C04-092B

80 引脚塑封薄型四方扁平封装（PT）—— 主体 12 x 12 x 1 mm，2.00 mm [TQFP]

注：最新封装图请至 http://www.microchip.com/packaging 查看 Mcirochip 封装规范。



建议的焊盘布局

单位		毫米		
尺寸范围		最小	正常	最大
触点间距	E		0.50 BSC	
触点焊盘间距	C1		13.40	
触点焊盘间距	C2		13.40	
触点焊盘宽度（X80）	X1			0.30
触点焊盘长度（X80）	Y1			1.50
焊盘间距	G	0.20		

注：

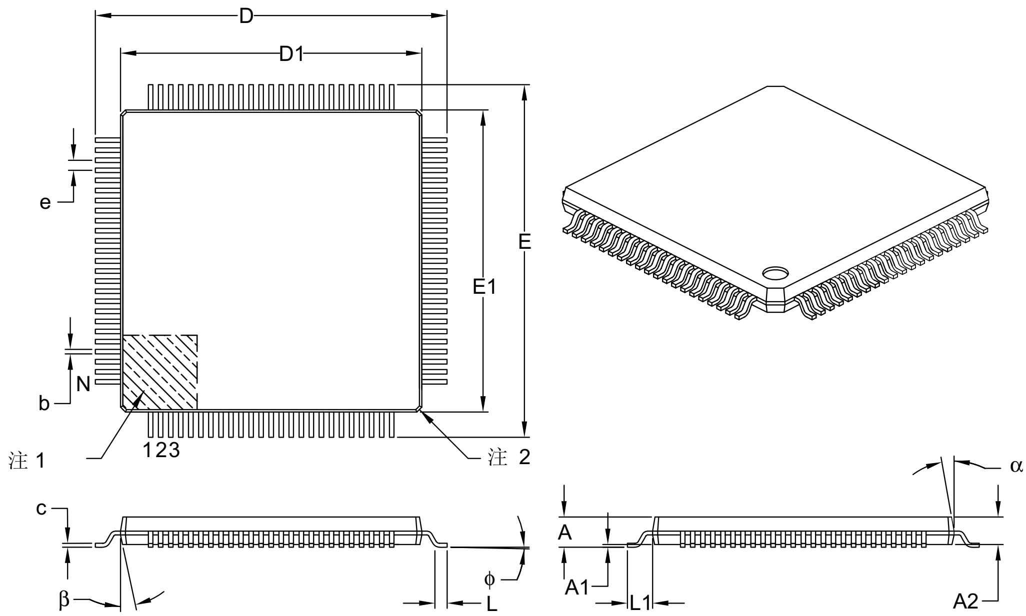
1. 尺寸和公差遵循ASME Y14.5M

BSC：基本尺寸。理论精确值，不含公差。

Microchip Technology 图号 C04-2092A

100 引脚塑封薄型四方扁平封装（PT）—— 主体 12 x 12 x 1 mm，2.00 mm [TQFP]

注：最新封装图请至 http://www.microchip.com/packaging 查看 Mcirochip 封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N	100		
引脚间距	e	0.40 BSC		
总高度	A	–	–	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	–	0.15
底脚长度	L	0.45	0.60	0.75
底脚占位距离	L1	1.00 REF		
底脚倾斜角	φ	0°	3.5°	7°
总宽度	E	14.00 BSC		
总长度	D	14.00 BSC		
塑模封装宽度	E1	12.00 BSC		
塑模封装长度	D1	12.00 BSC		
引脚厚度	c	0.09	–	0.20
引脚宽度	b	0.13	0.18	0.23
塑模顶部锥度	α	11°	12°	13°
塑模底部锥度	β	11°	12°	13°

注：

1. 引脚1定位标记可能会有变化，但必须位于阴影区域内。
2. 拐角处可能存在斜面，尺寸可变。
3. 尺寸D1和E1不包括塑模毛边或突起，塑模每侧的毛边或突起不得超过0.25 mm。
4. 尺寸和公差遵循ASME Y14.5M。

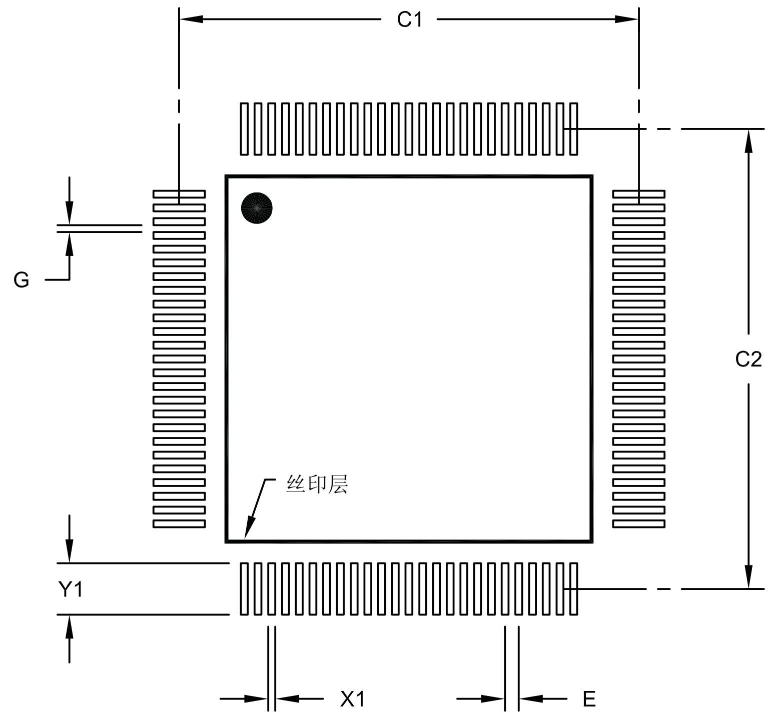
   BSC：基本尺寸。理论精确值，不含公差。

   REF：参考尺寸。通常不包含公差，仅供参考。

Microchip Technology 图号C04-100B

100 引脚塑封薄型四方扁平封装（PT）—— 主体 12 x 12 x 1 mm，2.00 mm [TQFP]

注：最新封装图请至 http://www.microchip.com/packaging 查看 Mcirochip 封装规范。



建议的焊盘布局

单位		毫米		
尺寸范围		最小	正常	最大
触点间距	E		0.40 BSC	
触点焊盘间距	C1		13.40	
触点焊盘间距	C2		13.40	
触点焊盘宽度（X100）	X1			0.20
触点焊盘长度（X100）	Y1			1.50
焊盘间距	G	0.20		

注：

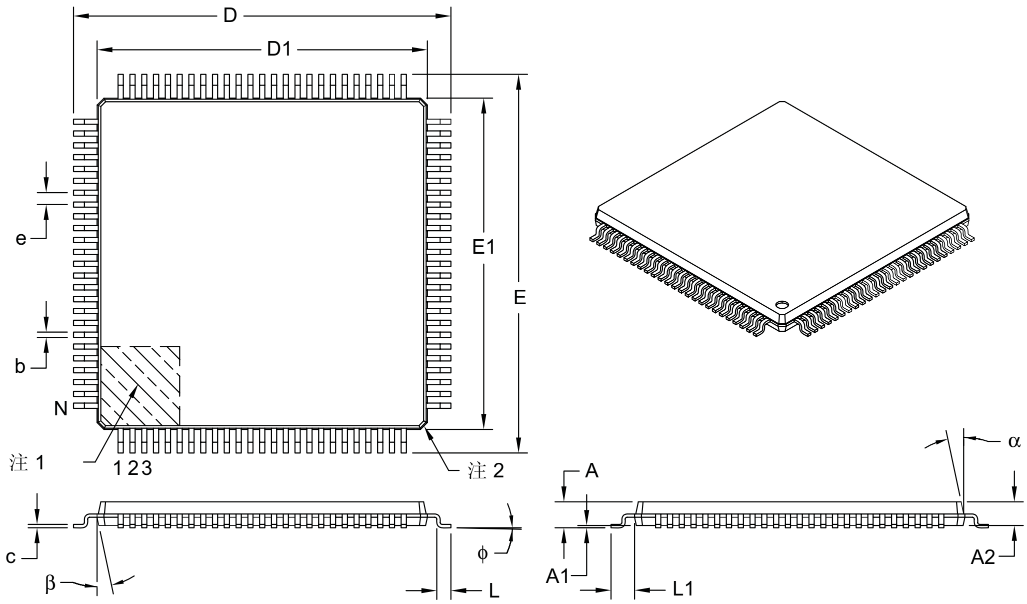
1. 尺寸和公差遵循ASME Y14.5M

BSC：基本尺寸。理论精确值，不含公差。

Microchip Technology 图号 C04-2100A

100 引脚塑封薄型四方扁平封装（PT）—— 主体 14 x 14 x 1 mm，2.00 mm [TQFP]

注：最新封装图请至 http://www.microchip.com/packaging 查看 Mcirochip 封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N		100	
引脚间距	e		0.50 BSC	
总高度	A	–	–	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	–	0.15
总长度	L	0.45	0.60	0.75
底脚占位距离	L1		1.00 REF	
底脚倾斜角	φ	0°	3.5°	7°
总宽度	E		16.00 BSC	
总长度	D		16.00 BSC	
塑模封装宽度	E1		14.00 BSC	
塑模封装长度	D1		14.00 BSC	
引脚厚度	c	0.09	–	0.20
引脚宽度	b	0.17	0.22	0.27
塑模顶部锥度	α	11°	12°	13°
塑模底部锥度	β	11°	12°	13°

注：

1. 引脚1的定位标记可能会有变化，但必须位于阴影区域内。
2. 拐角处可能存在斜面，尺寸可变。
3. 尺寸D1和E1不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过0.25 mm。
4. 尺寸和公差遵循ASME Y14.5M。

   BSC：基本尺寸。理论精确值，不含公差。

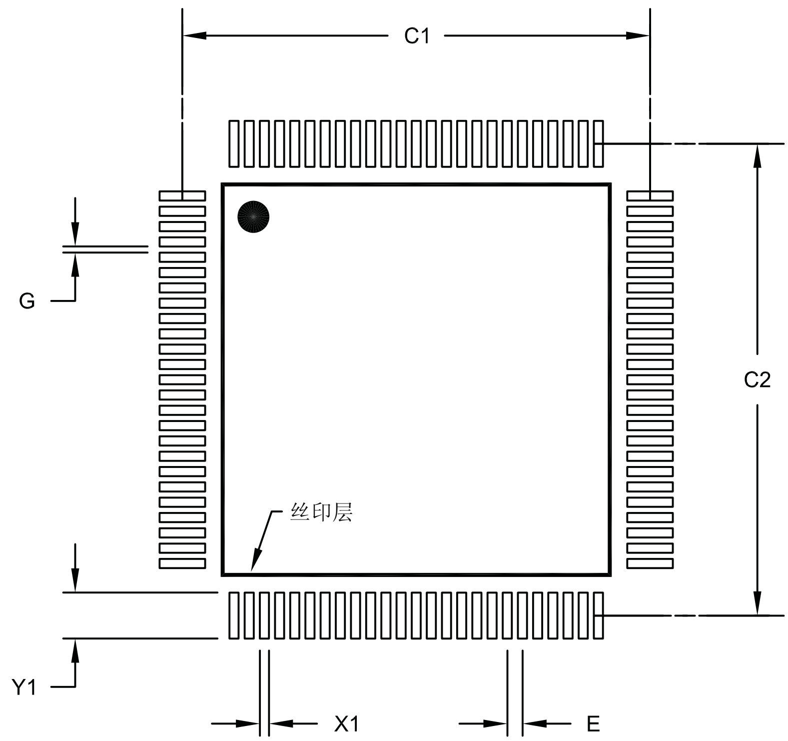
   REF：参考尺寸。通常不含有公差，仅供参考。

Microchip Technology 图号C04-110B

超前信息

100 引脚塑封薄型四方扁平封装（PT）—— 主体 14 x 14 x 1 mm，2.00 mm [TQFP]

注：最新封装图请至 http://www.microchip.com/packaging 查看 Mcirochip 封装规范。



建议的焊盘布局

单位		毫米		
尺寸范围		最小	正常	最大
触点间距	E		0.50 BSC	
触点焊盘间距	C1		15.40	
触点焊盘间距	C2		15.40	
触点焊盘宽度（X100）	X1			0.30
触点焊盘长度（X100）	Y1			1.50
焊盘间距	G	0.20		

注：

1. 尺寸和公差遵循ASME Y14.5M

BSC：基本尺寸。理论精确值，不含公差。

Microchip Technology图号 C04-2110A

注：

附录 A：　　　　版本历史

版本 A （2007 年 12 月）

PIC24FJ256GA110 系列器件数据手册的最初版本。

注：

索引

K

MICROCHIP 网站

Microchip 网站（www.microchip.com）为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的因特网浏览器即可访问。网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和样本程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及存档软件
- **一般技术支持**——常见问题（FAQ）、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 www.microchip.com，点击"变更通知客户（Customer Change Notification）"服务后按照注册说明完成注册。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师（FAE）
- 技术支持
- 开发系统信息热线

客户应联系其代理商、代表或应用工程师（FAE）寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 http：//support.microchip.com 获得网上技术支持。

读者反馈表

我们努力为您提供最佳文档，以确保您能够成功使用 Microchip 产品。如果您对文档的组织、条理性、主题及其他有助于提高文档质量的方面有任何意见或建议，请填写本反馈表并传真给我公司 TRC 经理，传真号码为 86-21-5407-5066。

请填写以下信息，并从下面各方面提出您对本文档的意见。

致: TRC 经理 总页数 ________

关于: 读者反馈

发自: 姓名 ________________________

公司 ________________________

地址 ________________________

国家 / 省份 / 城市 / 邮编 ________________________

电话: (______) ______________ 传真:(______) ______________

应用(选填):

您希望收到回复吗?是____ 否____

器件: PIC24FJ256GA110 系列 文献编号: DS39905A_CN

问题

1. 本文档中哪些部分最有特色?

____

____

2. 本文档是否满足了您的软硬件开发要求?如何满足的?

____

____

3. 您认为本文档的组织结构便于理解吗?如果不便于理解,那么问题何在?

____

____

4. 您认为本文档应该添加哪些内容以改善其结构和主题?

____

____

5. 您认为本文档中可以删减哪些内容,而又不会影响整体使用效果?

____

____

6. 本文档中是否存在错误或误导信息?如果存在,请指出是什么信息及其具体页数。

____

____

7. 您认为本文档还有哪些方面有待改进?

____

____

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或销售办事处联系。

PIC 24 FJ 256 GA1 10 T - I / PT - XXX

- **Microchip 商标**
- **架构**
- **闪存系列**
- **程序存储器容量（KB）**
- **产品组**
- **引脚数**
- **卷带标志（如果使用了话）**
- **温度范围**
- **封装**
- **模式**

架构	24	= 不包含 DSP 的 16 位改进哈佛架构
闪存系列	FJ	= 闪存程序存储器
产品组	GA1	= 通用单片机
引脚数	06 08 10	= 64 引脚 = 80 引脚 = 100 引脚
温度范围	I	= -40°C 到 +85°C （工业级）
封装	PF PT	= 100 引脚 （14x14x1 mm） TQFP （薄型四方扁平封装） = 64 引脚、 80 引脚和 100 引脚 （12x12x1 mm） TQFP （薄型四方扁平封装）
模式		三位数 QTP、 SQTP、代码或特殊要求 （空白为其他情况） ES = 工程样片

示例 ：

a) PIC24FJ128GA106-I/PT：
通用 PIC24F、 128 KB 程序存储器、 64 引脚、工业级温度和 TQFP 封装。

b) PIC24FJ256GA110-I/PT：
通用 PIC24F、 256 KB 程序存储器、 100 引脚、工业级温度和 TQFP 封装。

全球销售及服务网点

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277
技术支持：
http://support.microchip.com
网址：www.microchip.com

亚特兰大 **Atlanta**
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

波士顿 **Boston**
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 **Chicago**
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 **Dallas**
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 **Detroit**
Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

科科莫 **Kokomo**
Kokomo, IN
Tel: 1-765-864-8360
Fax: 1-765-864-8387

洛杉矶 **Los Angeles**
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣克拉拉 **Santa Clara**
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

加拿大多伦多 **Toronto**
Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 **Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 – 北京
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

中国 – 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 – 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 – 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 – 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 – 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 – 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 – 深圳
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 – 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 – 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 – 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 – 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

台湾地区 – 高雄
Tel: 886-7-536-4818
Fax: 886-7-536-4803

台湾地区 – 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 – 新竹
Tel: 886-3-572-9526
Fax: 886-3-572-6459

亚太地区

澳大利亚 **Australia - Sydney**
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 **India - Bangalore**
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

印度 **India - New Delhi**
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 **India - Pune**
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 **Japan - Yokohama**
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

韩国 **Korea - Daegu**
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 **Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 **Malaysia - Kuala Lumpur**
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 **Malaysia - Penang**
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 **Philippines - Manila**
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 **Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 **Thailand - Bangkok**
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 **Austria - Wels**
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 **Denmark-Copenhagen**
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 **France - Paris**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 **Germany - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 **Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 **Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

西班牙 **Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 **UK - Wokingham**
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/02/08